

Práce s textem - Návrh designu rozsáhlého dokumentu

Cíl: Navrhnou design profesionální publikace

Studijní materiály: \\NAS\Vyuka\PRV1

Zadání:

1. Text od 2. strany do konce vytiskněte
2. Vezměte si obyčejnou tužku a označte každý typ odstavce:
 - označte odstavec do rámečku nebo pomocí pruhu na boku odstavce
 - daný typ označte číslem vedle křížku (např #5)
 - u prvního odstavce daného typu uveďte navrhovaný název stylu
3. Na volné ploše papíru popište doporučený styl pro odstavec:
 - hlavička s názvem
 - formát znaků:
 - písmo (druh)
 - stupeň písma, barva, řez
 - formát odstavce:
 - zarovnání
 - odsazení
 - řádkování
 - ohraničení a pozadí
 - tok textu
4. Na jedné straně navrhnete formát strany:
 - nakreslete a okótujte okraje
 - nakreslete ukázkou záhlaví a zápatí
 - označte rozdílné sekce dokumentu

CVIČENÍ 1

NÁVRH DATABÁZE ODDÍL 1 - 3

ÚKOL #1

Napište věty ERDish pro každý vztah, popsany v modelu ERD společnosti DJ on Demand (ERD viz. příloha)

Write ERDish sentences for each relationship documented in the DJs on Demand ERD. Refer to Section 0, Database Design Instructor Course Resources.

Each CLIENT may be owner of one or more EVENTS.

Each EVENT must be owned by exactly one CLIENT

Each EVENT must be hold at PRIVATE HOME or PUBLIC SPACE.

Each PRIVATE HOME may be venue for one or more EVENTS.

Each PUBLIC SPACE may be venue for one or more EVENTS.

Each THEME may be catogorize one or more EVENTS.

Each EVENT may be categorized by exactly one THEME.

Each EVENT may be the source of one or more JOB ASSIGNMENTS.

Each JOB ASSIGNMENT must be for of exactly one EVENT.

Each PARTNER may be responsible for one or more JOB ASSIGNMENTS.

Each JOB ASSIGNMENT must be for one PARTNER.

Each MANAGER may be supervise one or more PARTNERS.

Each PARTNER may be supervised by exactly one MANAGER.

Each PLAY LIST ITEM must be for one SONG.

Each SONG may be an item on one or more PLAY LIST ITEMS.

Each EVENT may be require one or more PLAY LIST ITEMS.

Each PLAY LIST ITEM must be for one EVENT.

Each TYPE may be classify one or more SONGs.

Each SONG may be classified by exactly one TYPE.

Each SONG may be located on one or more TRACK LISTINGS.

Each TRACK LISTING must be locate one SONG.

Each EVENT must be classified by exactly one TYPE.

Each TYPE may be classify one or more EVENTS.

ÚKOL #2

Na základě následujícího scénáře určete entity a atributy. Nakreslete rámečky s entitami a seznamem atributů. Označte jedinečné atributy pomocí hash značky (#), a pokuste se určit volitelnost každého atributu („NULL“).

Given the following scenario, choose the entities and attributes. Draw the soft boxes with the entities and list the attributes underneath. Mark UNIQUE attributes with a hash mark (#), and try to determine optionality of each attribute.

Scénář: Moonlight Coffees is a fast growing chain of high quality coffee shops with currently over 500 shops in 12 countries of the world. Shops are located at first-class locations, such as major shopping, entertainment and business areas, airports, railway stations, and museums. Moonlight Coffees has some 9,000 employees.

All shops serve coffees, teas, soft drinks, and various kinds of pastries. Most shops sell nonfoods, like postcards and sometimes even theater tickets. Shop management reports sales figures on a daily basis to Headquarters, in local currency. Moonlight uses an internal exchange rates list that is changed monthly. Since January 1, 1999, the European Community countries must report in Euros.

ÚKOL #3

Zkopírujte a vložte scénář Summit Sporting (viz níže) do textového dokumentu a zvýrazněte všechna podstatná jména (entity a vztahy).

Copy and paste the Summit Sporting Goods scenario (below) into a text document and underline all nouns before the chat session. Have this ready to use during that chat.

Underlined – entity; marked - atribut

“I’m a manager of a sporting-goods wholesale company that operates worldwide to fill orders from retail sporting-goods stores. The stores are our customers (some of our people prefer to call them our clients). Right now we have 15 customers worldwide, but we’re trying to expand our customer base by about 10% each year starting this year. Our two biggest customers are Big John’s Sports Emporium in San Francisco and Womansports in Seattle. For each customer, we must track an ID and a name.

We may track an address (including the city, state, zip code, and country) and phone number. We maintain warehouses in different regions to best fill the order of our customers. For each order, we must track an ID. We may track the date ordered, date shipped, and payment type when the information is available.”

“Right now we have the world divided into five regions: North America, South America, Africa/Middle East, Asia, and Europe. That’s all we track; just the ID and name. We try to assign each customer to a region so we’ll generally know the best location from which to fill each order. Each warehouse must have an ID. We may track an address (including the city, state, zip code, and country) and phone number. We currently have only one warehouse per region, but we’re hoping to have more soon.

I manage the order-entry functions for our wholesale sporting-goods business. My department is responsible for placing and tracking the orders when our customers call. For each department, we must track the ID and name. Sometimes, our customers just mail us the orders when they are not in a rush, but most often they call us or fax us an order. We are hoping to expand our business by providing immediate turnaround of order information to our clients. Do you think we can put this application on the Web?”

“We can promise to ship by the next day as long as the goods are in stock (or inventory) at one of our warehouse locations. When the information is available, we track the amount in stock, the reorder point, maximum stock, a reason as to why we are out of stock, and the date we restocked the item. When the goods are shipped, we fax the shipping information automatically through our shipping system. No, I don’t manage that area. My department just ensures that our customers have the correct billing information and verifies that their account is in good credit standing. We may also record general comments about a customer.

We do make sure that all the items they have requested are in stock. For each item we track an ID. We may also track the item price, quantity, and quantity shipped if the information is available. If they are in stock, we want to process the order and tell our clients what the order ID is and how much their order total is. If the goods are not in stock, the customer tells us whether we should hold the order for a full shipment or process the partial order.”

“The accounting department is responsible for maintaining the customer information, especially for assigning new customer IDs. My department is allowed to update the customer information only when an order is placed and the billing or ship-to address has changed. No, we are not responsible for collections. That’s all handled by accounts receivable. I also think that the sales reps get involved because their commission depends on customers who pay! For each sales rep, or employee, we must know the ID and last name. Occasionally we need to know the first name, user ID, start date, title, and salary. We may also track the employee’s commission percent and any comments about the individual.

Our order-entry personnel are well versed in our product line. We hold frequent meetings with marketing so they can inform us of new products. This results in greater customer satisfaction because our order-entry operators can answer a lot of questions. This is possible because we deal with a few select customers and maintain a specialty product line. For each product, we must know the ID and name. Occasionally we must also know the description, suggested price, and unit of sale. We would also like the ability to track very long descriptions of our products and pictures of our

products, when it is necessary.”

CVIČENÍ 2

NÁVRH DATABÁZE

ODDÍL 5, PŘENOSITELNOST VZTAHU

NÁVRH DATABÁZE

ODDÍL 5, ŘEŠENÍ VZTAHU M:M

NÁVRH DATABÁZE

ODDÍL 6, NORMALIZACE DB

ÚKOL #1

Pro níže uvedené příklady nakreslete rámečky. Nakreslete vztahové linky a správně popište vztah v obou směrech. Kde je nutné vyznačte nepřenositelnost vztahu.

b. Do každého pokoje se může ubytovat jeden nebo více hostů. Každý host může být ubytován pouze v jednom pokoji.

d. V každém hotelu se může ubytovat jeden nebo více hostů. Každý host může být ubytován v jednom nebo více hotelích.

f. Každé oblečení musí mít pouze jednu cenu. Každá cena může být pro jedno nebo více oblečení.

h. Každý automobil musí používat pouze jeden rozměr pneumatik. Jedno nebo více aut může použít stejný rozměr pneumatik.

j. Každá osoba musí mít pouze jednu krevní skupinu. Každou krevní skupinu může mít jedna nebo více osob.

l. Každého studenta může učit jeden nebo více učitelů. Každý učitel může učit jednoho nebo více studentů.

n. Každý otisk prstu musí patřit pouze jedné osobě. Každá osoba musí mít pouze jeden otisk prstu.

Draw softboxes for each of the following. Draw relationship lines and correctly label each relationship in both directions. Indicate non-transferability when appropriate.

b. Each room may house one or more guests. Each guest may stay in one and only one room.

d. Each hotel may be the host of one or more guests. Each guest may be hosted in one or more hotels.

f. Each garment must have one and only one price. Each price may be for one or more garments.

h. Each automobile must use one and only one tire size. Each tire size may be used by one or more automobiles.

j. Each person must be of one and only one blood type. Each blood type may classify one or more

l. Each student may learn from one or more teachers. Each teacher may educate one or more students.

n. Each fingerprint must belong to one and only one person. Each person must have one and only one fingerprint.

ÚKOL #2

Vyřešte vztah M:M mezi učitelem a třídou, stejně jako mezi překladatelem a jazykem. Pro každou průnikovou entitu vymyslete další atributy jako UID.

Resolve the M:M between TEACHER and CLASS as well as INTERPRETER and LANGUAGE. For each intersection entity, think of additional attributes like a UID.

ÚKOL #3

Zkontrolujte, zda je každý ERD model je v 1NF. Pokud tomu tak není, proveďte potřebné změny. Check to see if each ERD is in 1NF. If not, make the necessary changes to correct it.

ÚKOL #4

Pomocí scénáře “Summit Sporting Goods”, který je popsán výše (Cvičení 1, úkol #3), nakreslete ERD s využitím entit a atributů projednaných na posledním chatu. Pokuste se vyřešit všechny M:N vztahy pomocí průnikových entit.

Using the “Summit Sporting Goods” scenario described above, draw an ERD using the entities and attributes discussed on your last chat. Try to resolve any many-to-many relationships with intersection entities.

CVIČENÍ 3

NÁVRH DATABÁZE

ODDÍL 6, NORMALIZACE DATABÁZE

NÁVRH DATABÁZE

ODDÍL 7, OBLOUKY

NÁVRH DATABÁZE

ODDÍL 7, MODELOVÁNÍ HISTORICKÝCH DAT

ÚKOL #1A

„Zápis do třídy“ je průnikovou entitou, která řeší vztah M:N mezi entitami STUDENT a CLASS. Jsou v následujícím ERD uplatněna pravidla 2NF? Pokud ne, opravte je.

Class Enrollment is the intersection entity that resolves the M:M between STUDENT and CLASS. Does the ERD follow the rules of Second Normal Form? If you spot a violation, correct it.

ÚKOL #1B

Lístek může být zakoupený od prodejce, v pokladně nebo na Internetu. Na lístku je zobrazen popis, událost, datum a cena. Prodejce má jméno a telefonní číslo. Pokladna má adresu a telefonní číslo. Internet má URL adresu.

Nakreslete entity a vyznačte exkluzivitu vztahu.

A show ticket is purchased from an agent, the box office, or the Internet. A ticket has a description, an event, a date and a price. An agent has a name and a phone number. The box office has an address and a phone number. The Internet has a URL address.

Draw the entities and represent the exclusive relationship.

ÚKOL #1C

Změňte ER model Video Store a doplňte následujícími požadavky:

Jak víte, potřebujeme zachovat historii všech našich výpůjček. Pokaždé, když si zákazník vypůjčí DVD, chceme uchovat datum a čas vypůjčení a datum a čas vrácení. Všechny naše DVD jsou splatné druhý den, takže nepotřebujeme uchovat datum splatnosti. Vedení historie výpůjček nám umožní analyzovat strukturu výpůjček. Budeme schopni určit kolik DVD si každý zákazník vypůjčil a kolikrát vrátil DVD pozdě. Budeme vědět, kolikrát bylo DVD vypůjčeno a tím pádem budeme vědět, kdy je nutné DVD vyřadit. Budeme vědět, které filmy preferují naši zákazníci.

Modify the Video Store ER model below to accommodate the following additional requirements: “You know, we really need to keep a history of all our rentals. Each time a customer rents a DVD, we would like to keep the rental date/time and the return date/time. All our DVDs are due back the next day, so we don’t need to keep a due date. Keeping this rental history will allow us to analyze

the pattern of our rentals. We will be able to determine how many DVDs each customer rents and how many times a customer has returned a DVD late. We will also know how many times a particular DVD has been used and will then know when to retire each DVD. We will also be able to analyze our customers' movie preferences.”

ÚKOL #2

Vytvořte hierarchický a rekurzivní model následující společnosti. Nakreslete pro každý z nich ERD.

Develop a hierarchical model and a recursive model for the following company scenario. Draw the ERD for each

Our company sells products throughout the World. So we've divided our company into four major sales regions: Region 1, Region 2, Region 3, and Region 4. Each sales region has a unique region code. Each sales region is then divided into sales districts. For example, Region 1 is divided into the U.S, Canadian, and Southern districts. Each district has a unique district code. Each district is made up of sales territories. The Southern District is composed of three territories: Mexico, South America, and U.S. Territories. The U.S. District is made up of three territories: the West, Middle, and East. The Canadian District is composed of two territories East and West. Each territory has a unique territory code.

Each sales territory is then broken down into sales areas. For example, South America is made up of two sales areas: Brazil, and the Coastal sales areas. Each sales area has a unique sales area code. The Brazil area includes Uruguay, Paraguay, and Ecuador.

Each salesperson is responsible for one or more sales areas, and has a specific sales quota. We also have sales managers who are responsible for one or more sales districts and sales directors who are responsible for one or more sales regions. Each sales manager is responsible for the territories with his districts. We don't overlap our employees' responsibilities. Sales area is always the responsibility of a single salesperson, and our managers and director's responsibilities don't overlap. Sometimes our salespersons, manager, and directors will be on leave or special assignments and will not have sales turf responsibilities. We identify all our sales personnel by their employee ids.

CVIČENÍ 4

NÁVRH DATABÁZE	ODDÍL 15, Anatomie SQL
NÁVRH DATABÁZE	ODDÍL 16, PRÁCE SE SLOUPCI A ŘÁDKY
NÁVRH DATABÁZE	ODDÍL 16, RELAČNÍ OPERÁTORY
NÁVRH DATABÁZE	ODDÍL 17, LOGICKÉ OPERÁTORY, PŘEDNOSTI
NÁVRH DATABÁZE	ODDÍL 17, TRÍDĚNÍ ŘÁDKŮ

Spustěte následující příkazy v Oracle Application Express. Vložte kopii každého dotaz do dokumentu Wordu nebo do Poznámkového bloku.

ÚKOL #1

Napište dotaz, který zobrazí příjmení a emailové adresy všech lidí z tabulky d_client databáze DJ on Demand. Hlavička sloupce by měla být "Client" nebo "Email Address.”

Write a query that displays the last_name and email addresses for all the people in the DJ on Demand d_client table. The column headings should appear as "Client" and "Email Address.”

```
select last_name "Client", email "Email Address"  
from d_clients;
```


ÚKOL #2

Ředitel Global Fast Foods se rozhodl dát všem zaměstnancům 5% navýšení hodinové mzdy a bonus \$.50 za hodinu. Nicméně když se díval na výsledky, nemohl pochopit, proč nové navýšení není takové, jaké předpokládal. Paní Doe by měla mít nový plat \$7.59, pan Miller by měl mít \$11.00 a Monique Tuttle by měla mít \$63.50. Použil následující dotaz. Co měl udělat?

The manager of Global Fast Foods decided to give all employees at 5%/hour raise + a \$.50 bonus/hour. However, when he looked at the results, he couldn't figure out why the new raises were not as he predicted. Ms. Doe should have a new salary of \$7.59, Mr. Miller's salary should be \$11.00, and Monique Tuttle should be \$63.50. He used the following query. What should he have done?

```
SELECT last_name, salary *1.05 +.50
FROM f_staffs;
```

ÚKOL #3

Ředitel Global Fast Foods by rád poslal kupóny pro následující prodej. Chce poslat jeden kupón do každé domácnosti. Vytvořte SELECT příkaz, který vrátí zákazníkovo příjmení a emailovou adresu.

The manager of Global Fast Foods would like to send out coupons for the upcoming sale. He wants to send one coupon to each household. Create the SELECT statement that returns the customer last name and a mailing address.

```
select last_name, address, city, state, zip
from f_customers;
```

ÚKOL #4

Sue, Bob a Monique byli zaměstnanci měsíce. Užítím tabulky f_staffs vytvořte SELECT příkaz, který zobrazí výsledky znázorněné v tabulce Super Star.

Sue, Bob, and Monique were the employees of the month. Using the f_staffs table, create a SELECT statement to display the results as shown in the Super Star chart.

```
select '*** ' || first_name || ' *** ' || first_name || ' ***' "Super Star"
from f_staffs;
```

ÚKOL #5

Global Fast Foods se rozhodl navýšit plat všem zaměstnancům o 5% . Vypracujte zprávu, která zobrazí výstup uvedený v tabulce.

Global Fast Foods has decided to give all staff members a 5% raise. Prepare a report that presents the output as shown in the chart.

```
select last_name "EMPLOYEE LAST NAME",
salary "CURRENT SALARY", salary*1.05 "SALARY WITH 5% RAISE"
from f_staffs;
```

ÚKOL #6

Majitelé DJs on Demand by rádi získali zprávu o všech položkách v tabulce D_CDs s následujícími hlavičkami sloupců: Inventory Item, CD Title, Music Producer a Year Purchased. Připravte tuto zprávu.

The owners of DJs on Demand would like a report of all items in their D_CDs table with the following column headings: Inventory Item, CD Title, Music Producer, and Year Purchased.

Prepare this report.

```
select CD_NUMBER "Inventory Item", TITLE "CD Title",
PRODUCER "Music Producer", YEAR "Year Purchased"
from d_cds;
```

ÚKOL #7

Užitím databáze Global Fast Foods získejte jméno, příjmení a adresu zákazníka, jehož ID je 456.

Using the Global Fast Foods database, retrieve the customer's first name, last name, and address for

the customer who uses ID 456.
select first_name, last_name, address
from f_customers
where ID = 456;

ÚKOL #8

Zobrazte název, datum začátku a konce propagační akce Global Fast Foods, kde se rozdávaly dárky "ballpen and highlighter".
Show the name, start date, and end date for Global Fast Foods' promotional item "ballpen and highlighter" giveaway.

```
select name, start_date, end_date  
from f_promotional_menus  
where give_away = 'ballpen and highlighter';
```

ÚKOL #9

Manažér DJ on Demand požaduje seznam všech CD titulů a roků výroby těch CD, které byly vyrobeny před rokem 2000.

The manager of DJ on Demand would like a report of all the CD titles and years of CDs that were produced before 2000.

```
select title, year  
from d_cds  
where year < 2000;
```

ÚKOL #10

Napište SQL příkaz, který zobrazí číslo studenta (studentno), jehož hlavní předmět v tabulce students je tělocvik. Označte sloupec studentno titulkem Student Number.

Write a SQL statement that will display the student number (studentno) of any student who has a PE major in the table named students. Title the studentno column Student Number.

```
select studentno "Student Number"  
from students  
where major is not null;
```

ÚKOL #11

Napište příkaz SQL, který vypíše zaměstnance Global Fast Foods narozené před rokem 1980.

Write a SQL statement that lists the Global Fast Foods employees who were born before 1980.

```
select first_name, last_name  
from F_staffs  
where birthdate < '1.1.1980';
```

ÚKOL #12

Zobrazte jméno, příjmení a plat všech zaměstnanců Global Fast Foods, jejichž plat se pohybuje v rozmezí \$5.00 až \$10.00 za hodinu.

Display the first name, last name, and salary of all Global Fast Foods staff whose salary is between \$5.00 and \$10.00 per hour.

```
select first_name, last_name, salary  
from F_staffs  
where salary between 5 and 10;
```

ÚKOL #13

Pouze užitím operátorů menší než, rovná se, větší než přepište následující dotaz:

```
SELECT first_name, last_name  
FROM f_staffs  
WHERE salary BETWEEN 20.00 and 60.00;
```


Using only the less than, equal, or greater than operators, rewrite the following query:

```
SELECT first_name, last_name
FROM f_staffs
WHERE salary BETWEEN 20.00 and 60.00;
SELECT first_name, last_name
FROM f_staffs
WHERE salary > 20 and salary <= 60;
```

ÚKOL #14

Vyberte všechny zaměstnance Oracle databáze, jejichž příjmení končí „s“. Změňte záhlaví sloupce na Possible Candidates.

Select all the Oracle database employees whose last names end with “s” Change the heading of the column to read Possible Candidates.

```
select last_name "Possible Candidates"
from employees
where last_name like '%s';
```

ÚKOL #15

Napište SQL příkaz, který vypíše seznam skladeb z inventáře DJs on Demand, jejichž typ kódu je 77, 12 nebo 1.

Write a SQL statement that lists the songs in the DJs on Demand inventory that are type code 77, 12 or 1

```
select title, type_code
from d_songs
where type_code IN (77, 12, 1);
```

ÚKOL #16

Zobrazte příjmení všech zaměstnanců Global Fast Foods, kteří mají „e“ a „i“ ve svém příjmení.

Display the last names of all Global Fast Foods employees who have “e” and “i” in their last names.

```
select last_name
from f_staffs
where last_name like '%e%' and last_name like '%i%';
```

ÚKOL #17

Pomocí tabulky zaměstnanci napište dotaz, který zobrazí všechny zaměstnance, jejichž příjmení začíná na „D“ a mají „a“ a „e“ kdekoli v příjmení.

Using the employees table, write a query to display all employees whose last names start with “D” and have “a” and “e” anywhere in their last name.

```
select last_name
from employees
where last_name like 'D%' and last_name like '%e%' and
last_name like '%a%';
```

ÚKOL #18

Kde jinde, než v soukromých domech pořádali DJs on Demand akce?

In which venues did DJs on Demand have events that were not in private homes?

```
select loc_type
from d_venues
where loc_type != 'Private Home';
```

ÚKOL #19

Kdo jsem?

Byl jsem přijat Oraclem po květnu 1998 ale před červnem 1999. Můj plat je menší než \$8000 za rok a mám „en“ v příjmení.

Who am I?

I was hired by Oracle after May 1998 but before June of 1999. My salary is less than \$8000 a year

and I have an “en” in my last name.
 select first_name, last_name
 from employees
 where hire_date between '31.5.1998' and '1.6.1999'
 and salary*12 < 8000 and last_name like '%en%';

ÚKOL #20

V níže uvedeném příkladu přejmenujte sloupec employee_id pomocí alias na „Number“. Dokončete příkaz SQL tak, aby seřadil výsledky podle sloupce s aliasem.

In the example below, assign the employee_id column the alias of “Number.” Complete the SQL statement to order the results set by the column alias.

```
SELECT employee_id as Numbers, first_name, last_name
FROM employees
order by Numbers;
```

ÚKOL #21

Seřďte písně databáze DJ on Demand sestupně podle názvu. Použijte alias „Our Collection“ pro název skladby.

Order the DJ on Demand songs by descending title. Use the alias “Our Collection” for the song title.

```
select title "Our Collection"
from d_songs
order by title desc;
```

ÚKOL #22

Napište SQL příkaz pomocí klauzule ORDER BY, který získá potřebné informace. Dotaz nespouštějte.

Vytvořte seznam studentů, kteří jsou v prvním ročníku školy. Zahrňte jméno, příjmení, ID studenta a číslo parkovacího místa. Seřadte výsledky abecedně podle příjmení studentů a následně podle jména. Jestliže více jak jeden student má stejné příjmení, seřadte jejich jména od Z do A. Všechny ostatní výsledky by měly být v abecedním pořadí (od A do Z).

Write a SQL statement using the ORDER BY clause that could retrieve the information needed. Do not run the query.

Create a list of students who are in their first year of school. Include the first name, last name, student ID number, and parking place number. Sort the results alphabetically by student last name and then by first name. If more than one student has the same last name, sort each first name in Z to A order. All other results should be in alphabetical order (A to Z).

```
select first_name, last_name, student_ID, parking_place
from students
order by last_name, first_name desc;
```

CVIČENÍ 5

PROGRAMOVÁNÍ SQL	ODDÍL 1, MANIPULACE SE ZNAKY
PROGRAMOVÁNÍ SQL	ODDÍL 1, ČÍSELNÉ FUNKCE
PROGRAMOVÁNÍ SQL	ODDÍL 1, DATUMOVÉ FUNKCE
PROGRAMOVÁNÍ SQL	ODDÍL 2, KONVERZNÍ FUNKCE
PROGRAMOVÁNÍ SQL	ODDÍL 2, NULL FUNKCE
PROGRAMOVÁNÍ SQL	ODDÍL 1, PODMÍNĚNÉ VÝRAZY

ÚKOL #1

Tři samostatná slova “Oracle,” “Internet,” a “Academy,” použijte v příkazu, který zobrazí následující výstup:

```
The Best Class
Oracle Internet Academy
```

Using the three separate words “Oracle,” “Internet,” and “Academy,” use one command to produce the following output:

The Best Class

Oracle Internet Academy

```
select CONCAT('Oracle', CONCAT(' Internet', ' Academy'))
```

"The Best Class" from dual;

ÚKOL #2

Jaké je pozice znaku “I” v řetězci “Oracle Internet Academy”?

What’s the position of “I” in “Oracle Internet Academy”?

```
select INSTR('Oracle Internet Academy', 'I') as position from dual;
```

ÚKOL #3

Řetězec “Oracle Internet Academy” doplňte na následující výstup: Oracle\$\$\$Internet\$\$\$Academy

Starting with the string “Oracle Internet Academy”, pad the string to produce:

Oracle\$\$\$Internet\$\$\$Academy

```
select REPLACE('Oracle Internet Academy', ' ', '$$$') as replace_space from dual;
```

ÚKOL #4

Použitím parametru (substituční proměnné) pro jméno oddělení napište dotaz, který vypíše ID oddělení, jméno oddělení, ID umístění pro oddělení zadané v proměnné

the_department_of_your_choice. Použijte tabulku DEPARTMENTS. Poznámka: všechny substituční proměnné jsou brány jako znakové řetězce, tudíž apostrofy (' ') nejsou nutné.

Using a substitution variable for the department name, write a query listing department id, department name and location id for departments located in the_department_of_your_choice. Use the DEPARTMENTS table. Note: All substitution variables in OAE are treated as character strings, so no quotes (‘ ’) are needed.

```
SELECT department_id, department_name, location_id
```

```
FROM departments
```

```
WHERE department_name = :the_department_of_your_choice;
```

ÚKOL #5

Zobrazte Oracle databázi zaměstnanců - příjmení a plat těch, jejichž employee_id je mezi 100 a 102. Připojte třetí sloupec, který vydělí každý plat hodnotou 1,55 a zaokrouhlí výsledek na dvě desetinná místa.

Display Oracle database employee last_name and salary for employee_ids between 100 and 102.

Include a third column that divides each salary by 1.55 and rounds the result to two decimal places.

```
select last_name, salary, round(salary/1.55 ,2) as th_col
```

```
from employees
```

```
where employee_id between 100 and 102
```

ÚKOL #6

Zobrazte příjmení a plat těch zaměstnanců, kteří pracují v oddělení 80. Zvyšte každému z nich plat o 5,33 % a výsledek ořežte na dvě desetinná místa.

Display employee last_name and salary for those employees who work in department 80. Give each of them a raise of 5.33% and truncate the result to two decimal places.

```
select last_name, salary, trunc(salary*1.0533 ,2) as
```

```
new_salary
```

```
from employees
```

```
where department_id = 80;
```

ÚKOL #7

Vydělte plat každého zaměstnance třemi. Zobrazte příjmení a platy pouze těch zaměstnanců, jejichž plat je násobkem 3.

Divide each employee’s salary by 3. Display only those employees’ last names and salaries who

earn a salary that is a multiple of 3.

```
select last_name, salary
from employees
where mod(salary, 3) = 0;
```

ÚKOL #8

Zobrazte počet dní mezi začátkem minulých letních prázdnin a začátkem letošního školního roku. Předpokládejme, že měsíc má 30,5 dne. Nazvěte výstup „Days“.

Display the days between the start of last summer's school vacation break and the day school started this year. Assume 30.5 days per month. Name the output "Days."

```
select months_between('1.9.2009', '1.7.2009') * 30.5 "Days" from dual;
```

ÚKOL #9

Pomocí jednoho výrazu zaokrouhlete dnešní datum s přesností na měsíc a rok a také jej ořežte s přesností na měsíc a rok. Použijte alias pro každý sloupec.

Using one statement, round today's date to the nearest month and nearest year and truncate it to the nearest month and nearest year. Use an alias for each column.

```
select round(sysdate, 'MM') as today_round_MM,
round(sysdate, 'YYYY') as today_round_YYYY,
trunc(sysdate, 'MM') as today_trunc_MM,
trunc(sysdate, 'YYYY') as today_trunc_YYYY
from dual;
```

ÚKOL #10

Zobrazte počet let mezi dnem narození zaměstnance Boba Millera a dnešním dnem. Zaokrouhlete na nejbližší rok.

Display the number of years between the Global Fast Foods employee Bob Miller's birthday and today. Round to the nearest year.

```
select round((sysdate - birthdate)/365.25) as Age
from f_staffs
where first_name = 'Bob' and last_name = 'Miller';
```

ÚKOL #11

Učitel řekl, že termín odevzdání projektu je poslední den tohoto měsíce. Který den to bude? Nazvěte výstup "Deadline."

The teacher said you have until the last day of this month to turn in your research paper. What day will this be? Name the output, "Deadline."

```
select last_day(sysdate) as Deadline
from dual;
```

ÚKOL #12

Vypište příjmení a datum narození zaměstnanců Global Fast Food. Převed'te datum narození na znaková data ve formátu Month DD, YYYY.

List the last names and birthdays of Global Fast Food Employees. Convert the birth dates to character data in the Month DD, YYYY format. Suppress any leading zeros. Potlačte výpis úvodních nul.

```
select last_name,
to_char(birthdate, 'Month DD, YYYY') as birthdate
from f_staffs;
```

ÚKOL #13

Zformátujte dotaz z tabulky f_promotional_menus databáze Global Fast Foods pro tisk data zahájení propagace s kódem 110 ve tvaru: Propagace začala desátého února 2004.

Format a query from the Global Fast Foods f_promotional_menus table to print out the start_date of promotional code 110 as: The promotion began on the tenth of February 2004.

```
select 'The promotion began on the ' || to_char(start_date, 'ddspth') || ' of ' || to_char(start_date,
'Month YYYY') || ' ' as start_date
from f_promotional_menus
where code = 110;
```

ÚKOL #14

Ellen Abel je zaměstnankyní, které byl zvýšen plat o 2000 \$. Zobrazte její jméno a příjmení, současný plat a nový plat. Zobrazte oba platy se znakem \$ a dvěma desetinnými místy. Nazvěte sloupec nového platu jako New Salary.

Ellen Abel is an employee who has received a \$2,000 raise. Display her first name and last name, her current salary, and her new salary. Display both salaries with a \$ and two decimal places. Label her new salary column AS New Salary.

```
select first_name, last_name, to_char(salary, '$9999,999.00') as Salary, to_char(salary + 2000,
'$9999,999.00') as "New Salary"
from Employees
```

```
where first_name = 'Ellen' and last_name = 'Abel';
```

ÚKOL #15

Vytvořte dotaz, který zformátuje v tabulce d_packages sloupce low-range a high-range, kde náklady za balíček budou ve formátu \$2500.00.

Create a query that will format the DJ on Demand d_packages columns, low-range and high-range package costs, in the format \$2500.00.

```
select to_char(low_range, '$9999,999.00') as "low range",
to_char(high_range, '$9999,999.00') as "high range"
from d_packages;
```

ÚKOL # 16

vytvořte zprávu, která zobrazí v databázi Global Fast Foods název propagace, datum zahájení a ukončení z tabulky f_promotional_menus. Pokud je uvedeno datum ukončení, dočasně ho nahraďte výrazem “end in two weeks.” Jestliže není uvedeno žádné datum ukončení, nahraďte jej dnešním datem.

Create a report that shows the Global Fast Foods promotional name, start date and end date from the f_promotional_menus table. If there is an end date, temporarily replace it with “end in two weeks.” If there is no end date, replace it with today’s date.

```
select name, NVL2(end_date, 'end in two weeks',
to_char(sysdate, 'DD.MM.YY'))
from F_PROMOTIONAL_MENUS;
```

ÚKOL #17

Manažér Global Fast Foods se rozhodl dát všem zaměstnancům, kteří nevydělávají přesčas přesčasový příplatek \$5.00. Vytvořte dotaz, který zobrazí příjmení a přesčasový příplatek \$5.00.

The manager of Global Fast Foods has decided to give all staff that currently does not earn overtime an overtime rate of \$5.00. Construct a query that displays last names and overtime rate shown as \$5.00.

```
select last_name,
NVL(to_char(overtime_rate), '$5.00') as "Over time"
from f_staffs;
```

ÚKOL # 18

Všechny null hodnoty ve sloupci specialty z tabulky d_partners databáze DJs on Demand nahraďte výrazem “No Specialty.” Zobrazte jen jméno a sloupec specialty.

For all null values in the specialty column in the DJs on Demand d_partners table, substitute “No Specialty.” Show the first name and specialty columns only.

```
select first_name, NVL(specialty, 'No Specialty') as
Specialty
```

from d_partners;

ÚKOL #19

Vytvořte dotaz z tabulky d_songs databáze DJ on Demand, který nahradí 2-minutové písně výrazem “shortest” a 10-minutové písně výrazem “longest.” Označte sloupec s výstupem “Play Times.”

From the DJ on Demand d_songs table, create a query that replaces the 2-minute songs with “shortest” and the 10-minute songs with “longest.” Label the output column “Play Times.”

```
select title,  
DECODE(duration,  
    '2 min', 'shortest',  
    '10 min', 'longest') as "Play Times"
```

from d_songs;

ÚKOL #20

Použijte tabulku employees Oracle databáze a CASE výraz pro dekódování ID oddělení. Zobrazte ID oddělení, příjmení, plat a sloupec nazvaný “New Salary”, jehož hodnoty jsou založeny na následujících podmínkách:

Jestliže ID oddělení je 10 potom plat vynásobte 1,25

Jestliže ID oddělení je 90 potom plat vynásobte 1,5

Jestliže ID oddělení je 130 potom plat vynásobte 1,75

Jinak zobrazte původní plat.

Use the Oracle database employees table and CASE expression to decode the department id. Display the department id, last name, salary and a column called “New Salary” whose value is based on the following conditions:

If the department id is 10 then 1.25 * salary

If the department id is 90 then 1.5 * salary

If the department id is 130 then 1.75 * salary

Otherwise, display the old salary.

```
select department_id, last_name, salary,
```

```
CASE department_id
```

```
  WHEN 10 then 1.25 * salary
```

```
  WHEN 90 then 1.5 * salary
```

```
  WHEN 130 then 1.75 * salary
```

```
  ELSE salary
```

```
END AS "New Salary"
```

```
from employees;
```

ÚKOL #21

Zobrazte jméno, příjmení, ID manažera a provizi v procentech všech zaměstnanců v odděleních 80 a 90. Zobrazte ID manažera v dalším sloupci nazvaném “Review.” Jestliže nemají manažera, zobrazte procenta provize. Jestliže nemají provizi, zobrazte 99999.

Display the first name, last name, manager ID, and commission percentage of all employees in departments 80 and 90. Display the manager ID in an additional column called “Review.” If they don’t have a manager, display the commission percentage. If they don’t have a commission, display 99999.

```
select first_name, last_name, manager_ID, commission_pct,
```

```
CASE WHEN manager_id IS NULL THEN commission_pct
```

```
  ELSE COALESCE(commission_pct, 99999)
```

```
END as "Review"
```

```
from employees
```


where department_id in(80, 90);

CVIČENÍ 6

PROGRAMOVÁNÍ SQL ODDÍL 3, CROSS A NATURAL JOIN

PROGRAMOVÁNÍ SQL ODDÍL 3, KLAUZULE JOIN

PROGRAMOVÁNÍ SQL ODDÍL 3, INNER & OUTER JOIN

PROGRAMOVÁNÍ SQL ODDÍL 4, GROUP FUNCTION

PROGRAMOVÁNÍ SQL ODDÍL 4, COUNT, DISTINCT, NVL

ÚKOL #1

Vytvořte cross-join (křížové spojení), který zobrazí příjmení a název oddělení z tabulek employees a departments.

Create a cross-join that displays the last name and department name from the employees and departments tables.

```
select last_name, department_name
from departments cross join employees;
```

ÚKOL #2

Vytvořte dotaz, kde použijete natural join pro spojení tabulek departments a locations pomocí sloupce location_id. Zobraz id oddělení, název oddělení, id umístění a město.

Create a query that uses a natural join to join the departments table and the locations table by the location_id column. Display the department id and name, location id, and city.

```
select department_id, department_name, location_id, city
from locations natural join departments;
```

ÚKOL #3

Vytvořte dotaz, kde použijete natural join pro spojení tabulky departments pomocí sloupce location_id. Omezte výstup pro id oddělení od 20 do 50. Zobrazte id oddělení, název oddělení, id umístění a město.

Create a query that uses a natural join to join the departments table by the location_id column. Restrict the output to only department IDs of 20 and 50. Display the department id and name, location id, and city.

```
select department_id, department_name, location_id, city
from locations natural join departments
where department_id IN(20, 50);
```

ÚKOL #4

Spojte v Oracle databázi tabulky locations a departments na sloupci location_id. Omezte výstup pouze pro umístění s číslem 1400

Join the Oracle database locations and departments table using the location_id column. Limit the results to location 1400 only.

```
select department_name, location_id
from locations join departments
using (location_id)
where location_id = 1400;
```

ÚKOL #5

Zobrazte název země, id oblasti a název oblasti pro americký kontinent. Vyberte region_ID, region_name, country_name.

Display country name, region ID and region name for Americas. Select region_ID, region_name, country_name

```
select region_ID, region_name, country_name
from regions join countries
using (region_id)
where region_name = 'Americas';
```

ÚKOL #6

Napište příkaz, který zobrazí ID zaměstnance, jméno, příjmení zaměstnance, ID manažera, jméno a příjmení manažera každého zaměstnance z tabulky employees. Tip: Je to self-join.

Write a statement that displays the employee ID, first name, last name, manager ID, manager first name, and manager last name for every employee in the employees table. Hint: this is a self-join.

```
select B.employee_ID,  
B.first_name,  
B.last_name,  
A.employee_ID as "manager ID",  
A.first_name as "manager first name",  
A.last_name as "manager last name"  
from employees A join employees B  
ON (A.employee_id = B.manager_id);
```

ÚKOL #7

Zobraz ID manažera, ID oddělení, jméno oddělení, jméno a příjmení všech zaměstnanců v oddělení 80, 90, 110 a 190.

Query and display manager ID, department ID, department name, first name, and last name for all employees in departments 80, 90, 110, and 190.

```
select manager_ID, department_ID, department_name,  
first_name, last_name  
from departments natural join employees  
where department_id IN (80, 90, 110, 190);
```

Display the employee's last name and employee number along with the manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

Upravte úkol 7 tak aby jste zobrazili všechny zaměstnance včetně těch, kteří nemají žádného manažera. Seřadte výsledky podle počtu zaměstnanců.

Modify problem 4 to display all employees, including those who have no manager. Order the results by the employee number

```
select B.employee_ID "Emp#",  
B.last_name "Employee",  
A.employee_ID as "Mgr#",  
A.last_name as "Manager"  
from employees A full outer join employees B  
ON (A.employee_id = B.manager_id)  
order by b.employee_id;
```

ÚKOL #8

Vytvořte dotaz, který zobrazí průměrnou cenu události databáze DJ on Demand. Zaokrouhlete na dvě desetinná místa.

Create a query that will show the average cost of the DJ on Demand events. Round to two decimal places.

```
select AVG(cost) "Average Cost"  
from d_events;
```

ÚKOL #9

Najděte průměrný plat zaměstnanců Global Fast Foods, jejichž ID manažera je 19.

Find the average salary for Global Fast Foods staff members whose manager ID is 19

```
select AVG(salary) "Average Salary"  
from f_staffs  
where manager_id = 19;
```

ÚKOL #10

Kolik písní je uvedeno v tabulce D_SONGS databáze DJs on Demand?

How many songs are listed in the DJs on Demand D_SONGS table?

```
select COUNT(*) "Count of songs"
from d_songs;
```

Na kolika různých místech měli DJs on Demand akci?

In how many different location types has DJs on Demand had venues?

```
select COUNT(distinct loc_type) "Count of type locations"
from d_venues;
```

CVIČENÍ 7

PROGRAMOVÁNÍ SQL ODDÍL 6, GROUP BY & HAVING

PROGRAMOVÁNÍ SQL ODDÍL 6, PODDOTAZY

PROGRAMOVÁNÍ SQL ODDÍL 6, JEDNOŘÁDKOVÉ PODDOTAZY

PROGRAMOVÁNÍ SQL ODDÍL 6, VÍCEŘÁDKOVÉ PODDOTAZY

ÚKOL #1

Každý z těchto SQL dotazů obsahuje chybu. Najděte ji a opravte. K ověření správnosti použijte Oracle Application Express.

Each of the following SQL queries has an error. Find the error and correct it. Use Oracle Application Express to verify that your corrections produce the desired results.

a:

```
SELECT manager_id, AVG(salary)
FROM employees
WHERE salary <16000
GROUP BY manager_id;
```

b:

```
SELECT cd_number, COUNT(title)
FROM d_cds
WHERE cd_number < 93;
```

c:

```
SELECT ID, MAX(ID), artist AS Artist
FROM d_songs
WHERE duration IN('3 min', '6 min', '10 min')
HAVING MAX(ID) < 50
GROUP by ID, artist;
```

d:

```
SELECT loc_type, rental_fee AS Fee
FROM d_venues
WHERE id <100
ORDER BY Fee;
```

ÚKOL #2

Které záznamy DJs on Demand z tabulky d_play_list_items mají stejné event_id jako ty, kde song_id je 45.

What DJs on Demand d_play_list_items song_id's have the same event_id as song_id 45?

```
select *
from d_play_list_items
where event_id = (select event_id
                  from d_play_list_items
                  where song_id = 45);
```

Které události v databázi DJs on Demand stojí více než ty, jejichž event_id = 100?

Which events in the DJs on Demand database cost more than event_id = 100?

```
select id, name
```

```
from d_events
where cost > (select cost
              from d_events
              where id = 100);
```

ÚKOL #3

Která pracovní pozice v Global Fast Foods má nižší plat než kterýkoliv kuchař?

What is the staff type for those Global Fast Foods jobs that have a salary less than those of any Cook staff-type jobs?

```
select staff_type
from f_staffs
where salary = (select MIN(Salary)
                from f_staffs);
```

ÚKOL #4

Najděte příjmení všech zaměstnanců, jejichž plat je stejný jako nejmenší mzda z každého oddělení.
Find the last names of all employees whose salaries are the same as the minimum salary for any department.

```
select last_name, salary
from employees
where salary = ANY (select min(salary)
                    from employees
                    group by department_id);
```

ÚKOL #5

Který zaměstnanec z Global Fast Foods má nejmenší plat? Tip: Můžete použít jak jednořádkový, tak víceřádkový poddotaz.

Which Global Fast Foods employee earns the lowest salary? Hint: You can use either a single-row or a multiple-row subquery.

var. A: Single-Row

```
select last_name, salary "Min Salary"
from employees
where salary = (select MIN(salary)
                from employees);
```

var. B: Multi-Row

```
select last_name, salary "Min Salary"
from employees
where salary <= ALL (select salary
                    from employees);
```

CVIČENÍ 8

PROGRAMOVÁNÍ SQL	ODDÍL 7, PŘÍKAZ INSERT
PROGRAMOVÁNÍ SQL	ODDÍL 7, PŘÍKAZ UPDATE, DELETE
PROGRAMOVÁNÍ SQL	ODDÍL 7, DEFAULT, MERGE, MULTI i.
PROGRAMOVÁNÍ SQL	ODDÍL 8, TVORBA TABULEK
PROGRAMOVÁNÍ SQL	ODDÍL 10, MODIFIKACE TABULEK

ÚKOL #1

V databázi DJs on Demand byly právě koupeny čtyři CD. Použitím příkazu INSERT přidejte každé CD do tabulky copy_d_cds. Po dokončení vkladu proveďte příkaz SELECT * pro ověření vaší práce.

DJs on Demand just purchased four new CDs. Use an explicit INSERT statement to add each CD to the copy_d_cds table. After completing the entries, execute a SELECT * statement to verify your work.

```

CREATE TABLE copy_d_cds
AS (select * from d_cds);

INSERT INTO copy_d_cds(cd_number, title, producer, year)
VALUES
(97, 'Celebrate the Day', 'R&B Inc.', 2003);

INSERT INTO copy_d_cds(cd_number, title, producer, year)
VALUES
(98, 'Holiday Tunes for All Ages', 'Tunes are Us', 2004);

INSERT INTO copy_d_cds(cd_number, title, producer, year)
VALUES
(99, 'Party Music', 'Old Town Records', 2004);

INSERT INTO copy_d_cds(cd_number, title, producer, year)
VALUES
(100, 'Best of Rock and Roll', 'Old Town Records', 2004);

SELECT * FROM copy_d_cds;

```

ÚKOL #2

V databázi DJs on Demand jsou očekávány dvě nové události. Jedna je podzimní fotbalová párty a druhá párty ve stylu šedesátých let. Na tyto akce klienti vyžadují písně uvedené v tabulce. Přidejte tyto písně do tabulky copy_d_songs užitím příkazu INSERT.

DJs on Demand has two new events coming up. One event is a fall football party and the other event is a sixties theme party. The DJs on Demand clients requested the songs shown in the table for their events. Add these songs to the copy_d_songs table using an implicit INSERT statement.

```

INSERT INTO copy_d_cds(cd_number, title)
VALUES(52, 'Sufing Summer');

INSERT INTO copy_d_cds(cd_number, title)
VALUES(53, 'Victory Victory');

```

```
SELECT * FROM copy_d_cds;
```

ÚKOL #3

Monique Tuttle, manažérka Global Fast Foods, poslala memorandum požadující okamžitou změnu cen. Cena jahodového koktejlu se zvýší z 3,59 dolarů na 3,75 dolarů a cena za hranolky se zvýší na 1,20 dolarů. Tyto změny proveďte v tabulce copy_f_food_items.

Monique Tuttle, the manager of Global Fast Foods, sent a memo requesting an immediate change in prices. The price for a strawberry shake will be raised from \$3.59 to \$3.75, and the price for fries will increase to \$1.20. Make these changes to the copy_f_food_items table.

```

CREATE TABLE copy_f_food_items
AS SELECT * FROM f_food_items;

UPDATE copy_f_food_items
SET price = 3.75
WHERE description = 'Strawberry Shake';

UPDATE copy_f_food_items

```

```
SET price = 1.2
WHERE description = 'Fries';
```

ÚKOL #4

Přidejte tyto nové zákazníky do tabulky copy_f_customers. Možná jste už přidali Katie Hernandez. Podaří se vám přidat všechny tyto záznamy?

Add the new customers shown below to the copy_f_customers table. You may already have added Katie Hernandez. Will you be able to add all these records successfully?

```
INSERT INTO copy_f_customers
VALUES (145, 'Katie', 'Hernandez', '92 Chico Way', 'Los Angeles', 'CA', 98008, '8586667641');
INSERT INTO copy_f_customers
VALUES (225, 'Daniel', 'Spode', '1923 Silverado', 'Denver', 'CO', 80219, '7193343523');
INSERT INTO copy_f_customers
VALUES (230, 'Adam', 'Zurn', '5 Admiral Way', 'Seattle', 'WA', NULL, '4258879009');
```

We cannot put NULL for a ZIP code because this column is defined as NOT NULL.

ÚKOL #5

Sue Doe, vynikající zaměstnankyní, byl zvýšen plat. Nyní bude placena stejně jako Bob Miller. Aktualizujte její záznam v tabulce copy_f_staffs.

Sue Doe has been an outstanding Global Foods staff member and has been given a salary raise. She will now be paid the same as Bob Miller. Update her record in copy_f_staffs.

```
UPDATE copy_f_staffs
SET salary = (SELECT salary
              FROM copy_f_Staffs
              WHERE CONCAT(first_name,last_name) = 'BobMiller'
              WHERE CONCAT(first_name,last_name) = 'SueDoe');
```

ÚKOL #6

Kdy chcete nastavit výchozí (DEFAULT) hodnotu?

When would you want a DEFAULT value?

This option prevents null values from entering the columns if a row is inserted without a specified value for the column.

Using default values also allows you to control where and when the default value should be applied.

ÚKOL #7

Doplňte popis tabulky GRADUATE CANDIDATE. Sloupec credits je cizím klíčem odkazujícím na požadovanou tabulku.

Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table.

Column Name
student_id
last_name
first_name
credits
graduation_date
Key Type
Yes

Nulls/Unique

No / No
No / No
No / No
Yes / No
FK Column

Yes

Datatype
NUMBER
VARCHAR2
VARCHAR2
NUMBER
DATE
Length
6
30
20
3

ÚKOL #8

Napište příkaz pro vytvoření tabulky grad_candidates.

Write the syntax to create the grad_candidates table.

```
CREATE TABLE grad_candidates
(student_id NUMBER(6) PRIMARY KEY,
last_name VARCHAR2(30) NOT NULL,
first_name VARCHAR2(20) NOT NULL,
credits NUMBER(3) FOREIGN KEY REFERENCES parent_table(credit_id),
graduation_date DATE);
```

ÚKOL #9

Ověřte vytvoření tabulky pomocí příkazu DESCRIBE.

Confirm creation of the table using DESCRIBE

```
DESC grad_candidates;
```

ÚKOL #10

Vložte do tabulky o_employees nový sloupec s názvem "Termination." Datový typ pro nový sloupec by měl být VARCHAR2. Nastavte výchozí hodnotu pro tento sloupec jako SYSDATE ve formátu: February 20th, 2003.

In your o_employees table, enter a new column called "Termination." The datatype for the new column should be VARCHAR2. Set the DEFAULT for this column as SYSDATE to appear as character data in the format: February 20th, 2003.

```
CREATE TABLE o_employees
AS SELECT * from employees;
```

```
CREATE TABLE o_jobs
```

```
AS SELECT * from jobs;
```

```
DESC o_employees;
```

```
ALTER TABLE o_employees
```

```
ADD (termination VARCHAR2(20) DEFAULT TO_CHAR(SYSDATE, 'Month DDTH YYYY'));
```

ÚKOL #11

Vytvořte nový sloupec v tabulce o_employees s názvem start_date. Použijte datový typ **TIMESTAMP WITH LOCAL TIME ZONE**.

Create a new column in the o_employees table called start_date. Use the **TIMESTAMP WITH LOCAL TIME ZONE** as the datatype.

```
ALTER TABLE o_employees
```

```
ADD (start_date TIMESTAMP WITH LOCAL TIME ZONE);
```

ÚKOL #12

Užitím příkazu **CREATE** or **REPLACE** vytvořte pohled s názvem view_copy_d_songs, který zobrazí všechny sloupce z tabulky copy_d_songs.

Use the **CREATE** or **REPLACE** option to create a view of all the columns in the copy_d_songs table called view_copy_d_songs.

```
CREATE TABLE copy_d_songs
```

```
AS SELECT * from d_songs;
```

```
CREATE OR REPLACE VIEW view_copy_d_songs
```

```
AS
```

```
SELECT * FROM copy_d_songs;
```