# A3-20163-DBS301 --- *Updated version Nov 24, 2016*

# Changes are shown in RED.

# If submitted today or earlier, just ignore the changes

# DUE week 12 Friday before MIDNIGHT

**Extended deadline:** Monday Week 13 December 5, 2016 before midnight.

After midnight the value of assignment is zero.

**DANGER: There is a test in week 13. Doing the assignment late or getting a lot of help from someone without being able to understand it yourself will cause problems on the test.**

1 Change the name of this file to a3-YourEmailid

Mail it back as a WORD attachment and not a PDF or image or ONEDRIVE or ….. etc

2 You can do this in groups, but remember if you don't do the work and the other members do it, you will likely fail the test and quite possibly the exam.

3 As a group member you are saying that you participated fairly as part of a group of and that you understand everything that was submitted.

Good luck

1 CREATE TABLES Question

**DIVISION**

|  |  |  |
| --- | --- | --- |
| **Column Name** | DIVISION\_ID | DIVISION\_NAME |
| **Key Type** | **PK** |  |
| **Null/Unique** |  | **NN, U** |
| **FK Table** |  |  |
| **FK Column** |  |  |
| **Validation** |  |  |
| **Datatype** | **NUMBER** | **VARCHAR** |
| **Length** | **3** | **25** |
| **Sample data** |  |  |
|  | **10** | **East Coast** |
|  | **20** | **Quebec** |
|  | **30** | **Ontario** |

WAREHOUSE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | WAREHOUSE\_ID | CITY | RATING | FOUND\_DATE | DIVISION\_ID |
| **Key Type** | **PK** |  | **CK** |  | **FK** |
| **Null/Unique** |  | **NN, U** |  | **NN** | **NN** |
| **FK Table** |  |  |  |  | **DIVISION** |
| **FK Column** |  |  |  |  | **DIVISION\_ID** |
| **Validation** |  |  | **A, B, C, D** |  |  |
| **Datatype** | **NUMBER** | **VARCHAR** | **CHAR** | **DATE** | **NUMBER** |
| **Length** | **3** | **15** | **1** |  | **3** |
| **Sample Data** | **1** | **Montreal** | **A** | **Current date** | **10** |
|  | **7** | **Fredericton** | **B** | **Current date** | **10** |
|  | **10** | **Toronto** | **A** | **Current date** | **30** |

SECTION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | WAREHOUSE\_ID | SECTION\_ID | DESCRIPTION | CAPACITY |
| **Key Type** | **PK, FK** | **PK** |  |  |
| **Null/Unique** |  |  | **NN** |  |
| **FK Table** | **WAREHOUSE** |  |  |  |
| **FK Column** | **WAREHOUSE\_ID** |  |  |  |
| **Datatype** | **NUMBER** | **NUMBER** | **VARCHAR** | **NUMBER**  NEW  Sample data |
| **Length** | **3** | **2** | **50** | **8** |
| **Sample data** | **1** | **1** | **Whse 1 Floor 1** | **2000** |
|  | **1** | **2** | **Whse 1 Floor 2** | **500** |
|  | **7** | **1** | **Whse 7 Floor 1** | **15000** |

1 (10 marks) Write the required SQL statements to create tables WAREHOUSE, DIVISION and SECTION.

Follow these general rules in the process:

A. Create all CHECK (incl. NOT NULL) and UNIQUE as column level constraints

Constraint names needed for CHECK constraints. The other constraints (NN and UK) do not need a name.

B. Create all PK and FK constraints at the table level and give them proper names.

PUT ANSWERS starting here

CREATE TABLE division(

division\_id NUMBER(3),

division\_name VARCHAR(25) NOT NULL UNIQUE,

CONSTRAINT div\_id\_pk PRIMARY KEY(division\_id)

);

DESC DIVISION;

Name Null Type

------------------------------ -------- ----------------------------------

DIVISION\_ID NOT NULL NUMBER(3)

DIVISION\_NAME NOT NULL VARCHAR2(25)

INSERT INTO DIVISION VALUES(10, 'East Coast');

INSERT INTO DIVISION VALUES(20, 'Quebec');

INSERT INTO DIVISION VALUES(30, 'Ontario');

SELECT \*

FROM DIVISION;

DIVISION\_ID DIVISION\_NAME

---------------------- -------------------------

10 East Coast

20 Quebec

30 Ontario

CREATE TABLE warehouse(

warehouse\_id NUMBER(3),

city VARCHAR(15) NOT NULL UNIQUE,

rating CHAR(1) CONSTRAINT ware\_rat\_ck CHECK(rating IN('A','B','C','D')),

found\_date DATE NOT NULL,

division\_id NUMBER(3) NOT NULL,

CONSTRAINT ware\_id\_pk PRIMARY KEY(warehouse\_id),

CONSTRAINT ware\_div\_fk FOREIGN KEY(division\_id)

REFERENCES division(division\_id)

);

DESC WAREHOUSE;

Name Null Type

------------------------------ -------- ----------------------------------

WAREHOUSE\_ID NOT NULL NUMBER(3)

CITY NOT NULL VARCHAR2(15)

RATING CHAR(1)

FOUND\_DATE NOT NULL DATE

DIVISION\_ID NOT NULL NUMBER(3)

INSERT INTO WAREHOUSE VALUES(1, 'Montreal', 'A', SYSDATE, 10);

INSERT INTO WAREHOUSE VALUES(7, 'Fredericton', 'B', SYSDATE, 10);

INSERT INTO WAREHOUSE VALUES(10, 'Toronto', 'A', SYSDATE, 30);

SELECT \*

FROM WAREHOUSE;

WAREHOUSE\_ID CITY RATING FOUND\_DATE DIVISION\_ID

----------------- --------------- ------ -------------------- ------------

1 Montreal A 16-11-24 10

7 Fredericton B 16-11-24 10

10 Toronto A 16-11-24 30

CREATE TABLE section(

warehouse\_id NUMBER(3),

section\_id NUMBER(2),

description VARCHAR(50) NOT NULL,

capacity NUMBER(8),

CONSTRAINT sec\_ware\_pk PRIMARY KEY(warehouse\_id, section\_id),

CONSTRAINT sec\_ware\_fk FOREIGN KEY(warehouse\_id)

REFERENCES warehouse(warehouse\_id)

);

DESC SECTION;

Name Null Type

------------------------------ -------- ----------------------------------

WAREHOUSE\_ID NOT NULL NUMBER(3)

SECTION\_ID NOT NULL NUMBER(2)

DESCRIPTION NOT NULL VARCHAR2(50)

CAPACITY NUMBER(8)

INSERT INTO SECTION VALUES(1, 1, 'Whse 1 Floor 1', 2000);

INSERT INTO SECTION VALUES(1, 2, 'Whse 1 Floor 2', 500);

INSERT INTO SECTION VALUES(7, 1, 'Whse 7 Floor 1', 15000);

SELECT \*

FROM SECTION;

WAREHOUSE\_ID SECTION\_ID DESCRIPTION CAPACITY

--------------- ------------ ----------------- ----------------------

1 1 Whse 1 Floor 1 2000

1 2 Whse 1 Floor 2 500

7 1 Whse 7 Floor 1 15000

**2** (3 marks) After creating all tables add column MGR\_ID to table SECTION as a FK column, that is related to the PK column EMPLOYEE\_ID in table EMPLOYEE

ALTER TABLE section

ADD (mgr\_id NUMBER(6,0),

CONSTRAINT sec\_mgr\_fk FOREIGN KEY(mgr\_id)

REFERENCES employees(employee\_id));

DESC SECTION;

Name Null Type

------------------------------ -------- ----------------------------------

WAREHOUSE\_ID NOT NULL NUMBER(3)

SECTION\_ID NOT NULL NUMBER(2)

DESCRIPTION NOT NULL VARCHAR2(50)

CAPACITY NUMBER(8)

MGR\_ID NUMBER(6)

UPDATE SECTION

SET mgr\_id = 200

WHERE warehouse\_id = 1;

UPDATE SECTION

SET mgr\_id = 124

WHERE warehouse\_id = 7;

SELECT \* FROM SECTION;

WAREHOUSE\_ID SECTION\_ID DESCRIPTION CAPACITY MGR\_ID

-------------- ------------ ---------------- ------------------ ----------

1 1 Whse 1 Floor 1 2000 200

1 2 Whse 1 Floor 2 500 200

7 1 Whse 7 Floor 1 15000 124

3 (3 marks) Modify the CHECK constraint on column RATING in table WAREHOUSE, so that it also may accept a new value F.

ALTER TABLE warehouse

DROP CONSTRAINT ware\_rat\_ck;

ALTER TABLE warehouse

ADD CONSTRAINT ware\_rat\_ck CHECK(rating IN('A','B','C','D','F'));

ALTER TABLE warehouse succeeded.

4 (3 marks) Create a new **Sequence** called **Whse\_id\_seq** that will generate unique numbers for PK values in table WAREHOUSE, so that the numbers start at 410 with the step of 10 and upper limit is 700 and will have NO values stored in the memory.

CREATE SEQUENCE Whse\_id\_seq

START WITH 410

INCREMENT BY 10

MAXVALUE 700

NOCACHE

NOCYCLE;

CREATE SEQUENCE succeeded.

5 (3 marks) Add new row to table WAREHOUSE by using this sequence for a city in Atlanta with unknown rating **and division 30.** You will assume today’s date as a foundation date. The date is to be entered automatically, meaning you cannot enter a specific date.

INSERT INTO warehouse (warehouse\_id, city, found\_date, division\_id)

VALUES(Whse\_id\_seq.nextval, 'Atlanta', SYSDATE, 30);

1 rows inserted

SELECT \* FROM WAREHOUSE;

WAREHOUSE\_ID CITY RATING FOUND\_DATE DIVISION\_ID

----------------- --------------- ------ -------------------- ------------

1 Montreal A 16-11-24 10

7 Fredericton B 16-11-24 10

10 Toronto A 16-11-24 30

410 Atlanta 16-11-24 30

6 (5 marks) Create table CITIES **from table LOCATIONS,** but only for location numbers less than 2000 (do NOT create this table from scratch). 🡪 You will have 5 to 18 rows

CREATE TABLE cities AS

SELECT \*

FROM locations

WHERE location\_id < 2000;

7 (2 marks) Issue command to show the structure of the table CITIES

DESC CITIES;

Name Null Type

------------------------------ -------- ----------------------------------

LOCATION\_ID NUMBER(4)

STREET\_ADDRESS VARCHAR2(40)

POSTAL\_CODE VARCHAR2(12)

CITY NOT NULL VARCHAR2(30)

STATE\_PROVINCE VARCHAR2(25)

COUNTRY\_ID CHAR(2)

8 (1 mark) Issue the SELECT command on cities and show result here.

SELECT \*  
FROM CITIES;

LOCATION\_ID STREET\_ADDRESS POSTAL\_CODE CITY STATE\_PROVINCE COUNTRY\_ID

----------- ----------- ------------ ----------- --------- ------------------- ----------

1000 1297 Via Cola di Rie 00989 Roma IT

1100 93091 Calle della Testa 10934 Venice IT

1200 2017 Shinjuku-ku 1689 Tokyo Tokyo Prefecture JP

1300 9450 Kamiya-cho 6823 Hiroshima JP

1400 2014 Jabberwocky Rd 26192 Southlake Texas US

1500 2011 Interiors Blvd 99236 South San Francisco California US

1600 2007 Zagora St 50090 South Brunswick New Jersey US

1700 2004 Charade Rd 98199 Seattle Washington US

1800 147 Spadina Ave M5V 2L7 Toronto Ontario CA

1900 6092 Boxwood St YSW 9T2 Whitehorse Yukon CA

9 (5 marks) Create a View called **WhsSec\_Man\_vu** that will display for each Warehouse\_id and Section\_id, the City, Division and manager’s Last\_name.

Alias for Last\_name should be LName and for Division should be Group.

CREATE VIEW WhsSec\_Man\_vu AS

SELECT w.warehouse\_id, s.section\_id, w.city,

w.division\_id "Group",

e.last\_name "LName"

FROM section s JOIN warehouse w

on s.warehouse\_id = w.warehouse\_id

JOIN departments d

ON d.manager\_id = s.mgr\_id

JOIN employees e

ON d.manager\_id = e.employee\_id;

SELECT \* FROM whssec\_man\_vu;

WAREHOUSE\_ID SECTION\_ID CITY Group LName

--------------- ---------------- --------------- ---------- ------------

1 2 Montreal 10 Whalen

1 1 Montreal 10 Whalen

7 1 Fredericton 10 Mourgos

10 (1 mark) What is the SELECT command to issue if in 2 months I want to test if a view was actually was created

SELECT SUBSTR(OBJECT\_NAME, 0, 15) "Object Name",

OBJECT\_TYPE, CREATED

FROM USER\_OBJECTS

WHERE OBJECT\_TYPE = 'VIEW'

AND CREATED > ADD\_MONTHS(SYSDATE, -2);

Object Name OBJECT\_TYPE CREATED

------------------------ ------------------- -------------------------

ALLDEPTS VIEW 16-11-15

ALLDEPTSUMM VIEW 16-11-15

ALLEMPS VIEW 16-11-15

CAN\_CITY\_VU VIEW 16-11-22

WHSSEC\_MAN\_VU VIEW 16-11-24

**11 (1 mark) If you want to modify the view what is the first line of the command**

CREATE OR REPLACE VIEW (view name) AS

12 Issue a SET operator to show the rows that were in LOCATIONS but not in CITIES

SELECT \*

FROM locations

MINUS

SELECT \*

FROM cities;

LOCATION\_ID STREET\_ADDRESS POSTAL\_CODE CITY STATE\_PROVINCE COUNTRY\_ID

---------------------- ----------------------------- ----------- ------------ --

2000 40-5-12 Laogianggen 190518 Beijing CN

2100 1298 Vileparle (E) 490231 Bombay Maharashtra IN

2200 12-98 Victoria Street 2901 Sydney New South Wales AU

2300 198 Clementi North 540198 Singapore SG

2400 8204 Arthur St London UK

2500 Magdalen Centre, The Oxford Science Park OX9 9ZB Oxford Oxford UK

2600 9702 Chester Road 09629850293 Stretford Manchester UK

2700 Schwanthalerstr. 7031 80925 Munich Bavaria DE

2800 Rua Frei Caneca 1360 01307-002 Sao Paulo Sao Paulo BR

2900 20 Rue des Corps-Saints 1730 Geneva Geneve CH

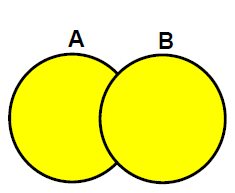
3000 Murtenstrasse 921 3095 Bern BE CH

3100 Pieter Breughelstraat 837 3029SK Utrecht Utrecht NL

3200 Mariano Escobedo 9991 11932 Mexico City Distrito Federal, MX

Using the following diagram as a hint and not a perfect representation.

Answer 13, 13, 15 and 16



13 All the rows in A and all the rows in B with no duplicates is the set operator called 🡺 **UNION**

14 All the rows in A and all the rows in B with duplicates 🡺 **UNION ALL**

15 The rows in common to BOTH A and B tables 🡺 **INTERSECT**

16 Rows that are in A but not in B would use the word 🡺 **MINUS**