Name: Shan Zhang

Student Id: 113004154

Email id: szhang146@myseneca.ca

Oracle id: dbs301\_171a31

Section : DBS301SAAL

# A3-20171-DBS301

# DUE week 12 Friday before MIDNIGHT

**Extended deadline:** for those who need it ……

THURSDAY Week 13 April 13, 2017 – no penalty if you take the extension

After midnight April 13 you will be really unlucky -- the value of the assignment is zero

🡺 meaning you lost 7%

**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 🡸🡸 must be done

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

2 In the subject line of the email put the file name from #1 above.

3 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.

4 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

List all members if any in your group.

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**  Sample data to insert |
| **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

-- division

create table division(

division\_id number(3),

division\_name varchar(25) not null unique,

constraint division\_division\_id\_pk primary key(division\_id)

);

insert into division values (10, 'East Coast');

insert into division values (20, 'Quebec');

insert into division values (30, 'Ontario');

Result:

DIVISION\_ID DIVISION\_NAME

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

10 East Coast

30 Ontario

20 Quebec

--warehouse

create table warehouse (

WAREHOUSE\_ID number(3),

CITY varchar(15) not null unique,

RATING char(1) constraint warehouse\_RATING\_ck check(RATING in ('A','B','C','D')),

FOUND\_DATE date not null,

DIVISION\_ID number(3) not null,

constraint warehouse\_WAREHOUSE\_ID\_pk primary key(WAREHOUSE\_ID),

constraint warehouse\_DIVISION\_ID\_fk foreign key(DIVISION\_ID) references division(division\_id)

);

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);

Result:

WAREHOUSE\_ID CITY RATING FOUND\_DATE DIVISION\_ID

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

1 Montreal A 08-APR-17 10

7 Fredericton B 08-APR-17 10

10 Toronto A 08-APR-17 30

--section

create table section (

WAREHOUSE\_ID number(3),

SECTION\_ID number(2),

DESCRIPTION varchar(50) not null,

CAPACITY number(8),

constraint section\_WHID\_SECID\_pk primary key(WAREHOUSE\_ID, SECTION\_ID),

constraint section\_WAREHOUSE\_ID\_fk foreign key(WAREHOUSE\_ID) references warehouse(WAREHOUSE\_ID)

);

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);

Result:

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));

alter table section

add constraint section\_mgr\_id\_fk foreign key(mgr\_id) references employees(employee\_id);

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 warehouse\_RATING\_ck;

alter table WAREHOUSE add constraint warehouse\_RATING\_ck check(RATING in ('A','B','C','D','F'));

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 420 with the step of 5 and upper limit is 700 and will have NO values stored in the memory.

create sequence Whse\_id\_seq

increment by 5

start with 420

maxvalue 700

nocache;

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

values(Whse\_id\_seq.nextval, 'Atlanta', NULL, sysdate, 30);

Result:

WAREHOUSE\_ID CITY RATING FOUND\_DATE DIVISION\_ID

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

1 Montreal A 08-APR-17 10

7 Fredericton B 08-APR-17 10

10 Toronto A 08-APR-17 30

420 Atlanta 08-APR-17 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

describe cities;

Result:

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;

Result:

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

10 rows selected

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 or replace view WhsSec\_Man\_vu

as

select w.warehouse\_id, s.section\_id, w.city, d.division\_name as "Group",

e.last\_name as "LName"

from warehouse w left join division d on (w.division\_id = d.division\_id)

full join section s on (w.warehouse\_id = s.warehouse\_id)

left join employees e on (s.mgr\_id = e.employee\_id);

select \* from WhsSec\_Man\_vu;

WAREHOUSE\_ID SECTION\_ID CITY Group LName

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

7 1 Fredericton East Coast

1 2 Montreal East Coast

1 1 Montreal East Coast

420 Atlanta Ontario

10 Toronto Ontario

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 view\_name, text from user\_views where view\_name=upper('view\_name');

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

create or replace view WhsSec\_Man\_vu as

(select …)

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

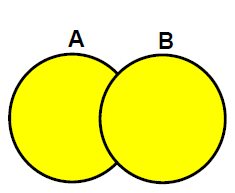
select \* from locations

minus

select \* from cities;

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

Answer 13, 14, 15 and 16



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

[Symbol]

UNION

14 All the rows in A and all the rows in B with duplicates [Symbol]

UNION ALL

15 The rows in common to BOTH A and B tables [Symbol]

INTERSECT

16 Rows that are in A but not in B would use the word [Symbol]

MINUS