Assignment 3 (7%)

Total : 60 marks.

Please work in **groups** of max 3 and min 2 to complete this Assignment. This Assignment is worth 7% of the total course grade and will be evaluated through your written submission, as well as the Assignment demo in the class worth 3%. During the Assignment demo, group members are randomly selected to present the answers to each of the lab questions. Group members not present during the Assignment demo will lose the demo mark. Individual submission will have penalty of 10%.

1. Add this declaration on the top of your file.

We, ------------(mention your names), declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

1. Specify what each member has done towards the completion of this work:

|  |  |  |
| --- | --- | --- |
|  | Name | Task(s) |
| 1 | ABIODUN OKE |  |
| 2 | SERGES DAMBOU |  |
| 3 | ADITI PATEL |  |
| 4 |  |  |

Solve the following using tables provided for labs like employees, departments, etc. Copy the commands under each question and also the screen shots of the result. For instance, if you create a view the result will be view is created. Take the screen shot and copy under the command. One member in the group upload the document on BlackBoard.

1. Display department number, department name, and the number of employees working in each department that has the highest number of employees. Use subquery only. (5 marks)

**ANSWER:**

**SELECT D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME, COUNT(E.EMPLOYEE\_ID)**

**FROM DEPARTMENTS D JOIN EMPLOYEES E**

**ON D.DEPARTMENT\_ID = E.DEPARTMENT\_ID**

**GROUP BY D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME**

**HAVING COUNT(E.EMPLOYEE\_ID)= (SELECT MAX( COUNT(E1.EMPLOYEE\_ID))**

**FROM EMPLOYEES E1**

**GROUP BY DEPARTMENT\_ID);**

**OUTPUT:**

**DEPARTMENT\_ID DEPARTMENT\_NAME COUNT(E.EMPLOYEE\_ID)**

**------------- ------------------------------ ---------------------------------------**

**50 Shipping 5**

1. Display all the employees who were hired on the day of the week on which the highest number of employees were hired. List the first name and last name columns together with a space in between. Label the column as Full Name and label day as day of the week. Use subquery only. (5 marks)

**ANSWER:**

**SELECT FIRST\_NAME||''||LAST\_NAME as "Full Name",to\_char(hire\_date,'Day') Day**

**FROM EMPLOYEES**

**WHERE to\_char(hire\_date,'Day') = (SELECT to\_char(hire\_date,'Day')**

**FROM EMPLOYEES**

**GROUP BY to\_char(hire\_date,'Day')**

**HAVING COUNT(\*)=(SELECT MAX(COUNT(\*))**

**FROM EMPLOYEES**

**GROUP BY to\_char(hire\_date,'Day')));**

**OUTPUT:**

**Full Name DAY**

**--------------------------------------------- ---------**

**BruceErnst Tuesday**

**KevinMourgos Tuesday**

**TrennaRajs Tuesday**

**JonathonTaylor Tuesday**

**ShelleyHiggins Tuesday**

**WilliamGietz Tuesday**

**6 rows selected**

1. Department 50 needs access to its employee data. Create a view named DEPT50 that contains the employee nuOmbers, employee last names, and department numbers for all employees in department 50. They have requested that you label the view columns EMPNO, EMPLOYEE, and DEPTNO. For security purposes, do not allow an employee to be reassigned to another department through the view. (5 marks)

**ANSWER:**

**CREATE VIEW DEPT50 AS**

**(SELECT EMPLOYEE\_ID "EMPNO", LAST\_NAME "EMPLOYEE", DEPARTMENT\_ID "DEPTNO"**

**FROM EMPLOYEES**

**WHERE DEPARTMENT\_ID = 50)**

**WITH CHECK OPTION CONSTRAINT EMP\_DEPT\_50;**

**OUTPUT:**

**EMPNO EMPLOYEE DEPTNO**

**---------- ------------------------- ----------**

**124 Mourgos 50**

**141 Rajs 50**

**142 Davies 50**

**143 Matos 50**

**144 Vargas 50**

1. Create a view named emp\_hired that has the names and hire dates for all employees who were hired before their managers, along with their manager’s names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively. (5 marks)

**ANSWER:**

**CREATE VIEW EMP\_HIRED AS**

**(SELECT E.FIRST\_NAME||' '||E.LAST\_NAME "Emloyee", E.HIRE\_DATE "Emp Hired",**

**M.MANAGER\_ID "Manager", M.HIRE\_DATE "Mgr Hired"**

**FROM EMPLOYEES E INNER JOIN EMPLOYEES M**

**ON E.MANAGER\_ID = M.EMPLOYEE\_ID**

**WHERE E.HIRE\_DATE < M.HIRE\_DATE**

**);**

**View EMP\_HIRED created.**

**OUTPUT:**

**Emloyee Emp Hired Manager Mgr Hired**

**---------------------------------------------- --------- ---------- ---------**

**Jennifer Whalen 87-09-17 100 89-09-21**

**Alexander Hunold 90-01-03 100 93-01-13**

**Trenna Rajs 95-10-17 100 99-11-16**

**Curtis Davies 97-01-29 100 99-11-16**

**Randall Matos 98-03-15 100 99-11-16**

**Peter Vargas 98-07-09 100 99-11-16**

**Ellen Abel 96-05-11 100 00-01-29**

**Jonathon Taylor 98-03-24 100 00-01-29**

**Kimberely Grant 99-05-24 100 00-01-29**

**9 rows selected**

1. The Date base Administrator needs you to create a table, which has a primary key constraint, but she wants to name the index to have a different name than the constraint. Create the Locations\_Index table based on the following table instance chart. Name the index for the PRIMARY KEY column as LOCATIONS\_PK\_IDX. (5 marks)

|  |  |  |
| --- | --- | --- |
| Column Name | Deptno | Dname |
| Primary key | Yes |  |
| Data Type | Number | VARCHAR2 |
| Length | 4 | 30 |

**ANSWER:**

**CREATE TABLE DEPT60 (**

**Deptno NUMBER(4) CONSTRAINT DEPT\_ID\_NO NOT NULL,**

**Dname VARCHAR2(30)**

**);**

**Table DEPT60 created.**

**CREATE INDEX LOCATION\_PK\_IDX ON DEPT60 (Deptno);**

**ALTER TABLE DEPT60**

**ADD CONSTRAINT LOCATION\_PK\_IDX PRIMARY KEY (Deptno);**

**Table DEPT60 altered**.

**OUTPUT:**

1. Create a report of a list of employees who are up for review in March; so you are requested to do the following: (5 marks)

Write a query to display the last names, month of the date of hire, and hire date of those employees who have been hired in the month of March, irrespective of the year of hire.

**ANSWER:**

**SELECT FIRST\_NAME||' '||LAST\_NAME, TO\_CHAR(HIRE\_DATE, 'MONTH'), HIRE\_DATE**

**FROM EMPLOYEES**

**WHERE TO\_CHAR(HIRE\_DATE, 'FMMONTH') = 'MARCH';**

**OUTPUT:**

**FIRST\_NAME||''||LAST\_NAME TO\_CHAR(H HIRE\_DATE**

**---------------------------------------------- --------- ---------**

**Randall Matos MARCH 98-03-15**

**Jonathon Taylor MARCH 98-03-24**

1. The Accounting department requires an analysis on maximum and minimum salaries by job, manager, and department. They have asked you to do the following: (5 marks)

Write a query to display the following groupings:

Department\_id, job\_id

Job\_id, manager\_id

**ANSWER:**

**SELECT DEPARTMENT\_ID, JOB\_ID, MANAGER\_ID, MAX(SALARY), MIN(SALARY)**

**FROM EMPLOYEES**

**GROUP BY**

**GROUPING SETS**

**( (DEPARTMENT\_ID, JOB\_ID),**

**(JOB\_ID, MANAGER\_ID)**

**);**

**OUTPUT:**

**DEPARTMENT\_ID JOB\_ID MANAGER\_ID MAX(SALARY) MIN(SALARY)**

**------------- ---------- ---------- --------------------------------------- ---------------------------------------**

**AC\_MGR 101 12000 12000**

**ST\_MAN 100 5800 5800**

**MK\_MAN 100 13000 13000**

**AD\_PRES 24000 24000**

**AD\_ASST 101 4400 4400**

**IT\_PROG 102 9000 9000**

**IT\_PROG 103 6000 4200**

**AC\_ACCOUNT 205 8300 8300**

**AD\_VP 100 17000 17000**

**ST\_CLERK 124 3500 2500**

**SA\_REP 149 11000 7000**

**SA\_MAN 100 10500 10500**

**MK\_REP 201 6000 6000**

**110 AC\_ACCOUNT 8300 8300**

**90 AD\_VP 17000 17000**

**50 ST\_CLERK 3500 2500**

**80 SA\_REP 11000 8600**

**50 ST\_MAN 5800 5800**

**80 SA\_MAN 10500 10500**

**110 AC\_MGR 12000 12000**

**90 AD\_PRES 24000 24000**

**60 IT\_PROG 9000 4200**

**20 MK\_MAN 13000 13000**

**SA\_REP 7000 7000**

**10 AD\_ASST 4400 4400**

**20 MK\_REP 6000 6000**

**26 rows selected**

1. Create the EMP table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and JOB\_ID columns for departments 90,80,60, and 50. Name the columns in your new table NO, FIRST\_NAME, LAST\_NAME, SALARY, and JOB\_TITLE, respectively. (5 marks)

**ANSWER:**

**CREATE TABLE EMP AS**

**(SELECT EMPLOYEE\_ID "NO", FIRST\_NAME, LAST\_NAME, SALARY, JOB\_ID "JOB\_TITLE"**

**FROM EMPLOYEES**

**WHERE DEPARTMENT\_ID IN (90, 80, 60, 50));**

**OUTPUT:**

**NO FIRST\_NAME LAST\_NAME SALARY JOB\_TITLE**

**---------- -------------------- ------------------------- ---------- ----------**

**124 Kevin Mourgos 5800 ST\_MAN**

**141 Trenna Rajs 3500 ST\_CLERK**

**142 Curtis Davies 3100 ST\_CLERK**

**143 Randall Matos 2600 ST\_CLERK**

**144 Peter Vargas 2500 ST\_CLERK**

**103 Alexander Hunold 9000 IT\_PROG**

**104 Bruce Ernst 6000 IT\_PROG**

**107 Diana Lorentz 4200 IT\_PROG**

**149 Eleni Zlotkey 10500 SA\_MAN**

**174 Ellen Abel 11000 SA\_REP**

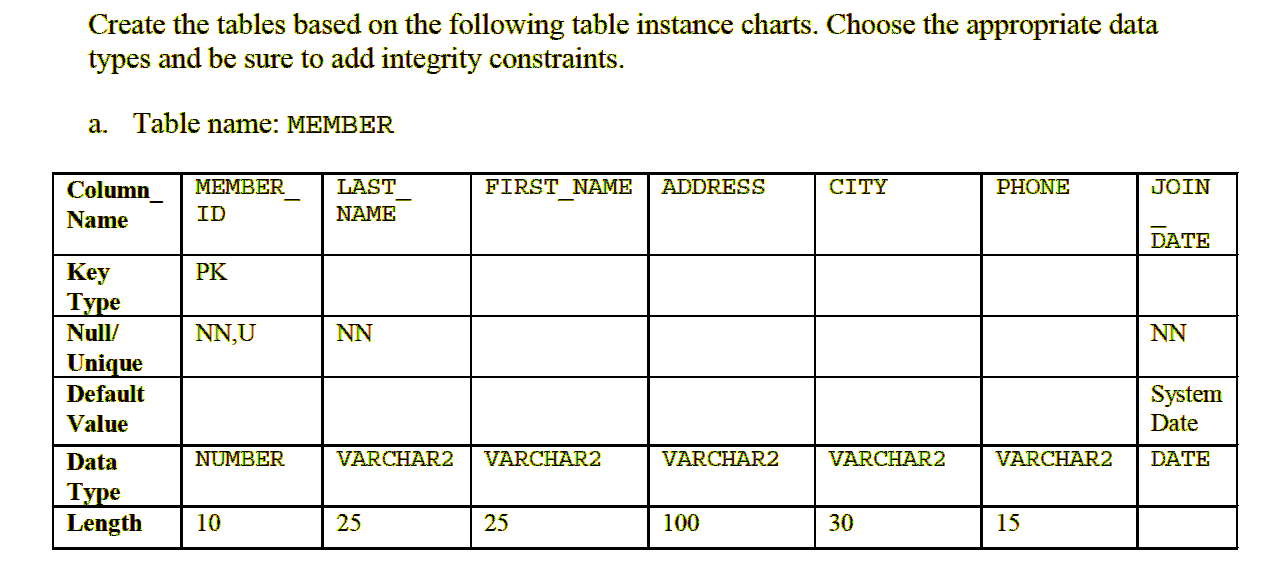
**176 Jonathon Taylor 8600 SA\_REP**

**100 Steven King 24000 AD\_PRES**

**101 Neena Kochhar 17000 AD\_VP**

**102 Lex De Haan 17000 AD\_VP**

1. (15 marks)



**ANSWER:**

**CREATE TABLE MEMBER (**

**MEMBER\_ID NUMBER(10) NOT NULL PRIMARY KEY,**

**LAST\_NAME VARCHAR2(25) NOT NULL,**

**FIRST\_NAME VARCHAR2(25),**

**ADDRESS VARCHAR2(100),**

**CITY VARCHAR2(30),**

**PHONE VARCHAR2(15),**

**JOIN\_DATE DATE DEFAULT (SYSDATE) NOT NULL );**

**OUTPUT:**

**Name Null Type**

**---------- -------- -------------**

**MEMBER\_ID NOT NULL NUMBER(10)**

**LAST\_NAME NOT NULL VARCHAR2(25)**

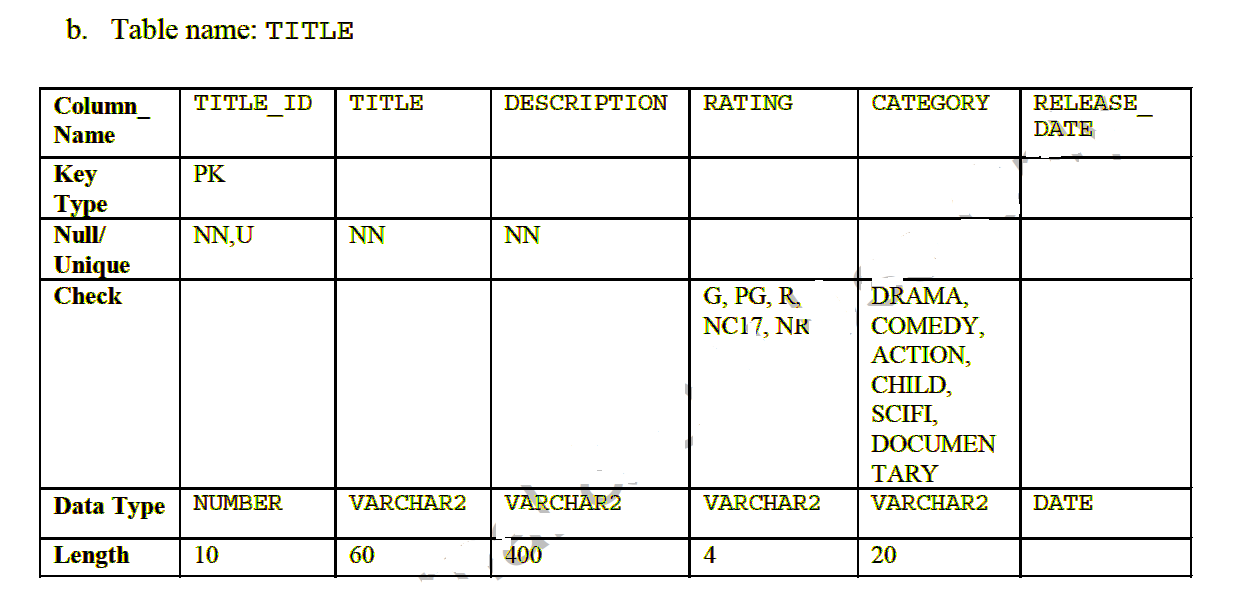
**FIRST\_NAME VARCHAR2(25)**

**ADDRESS VARCHAR2(100)**

**CITY VARCHAR2(30)**

**PHONE VARCHAR2(15)**

**JOIN\_DATE NOT NULL DATE**



**ANSWER:**

**CREATE TABLE TITLE**

**(**

**TITLE\_ID NUMBER(10) NOT NULL PRIMARY KEY,**

**TITLE VARCHAR2(60) NOT NULL ,**

**DESCRIPTION VARCHAR2(100) NOT NULL ,**

**RATING VARCHAR2(4) CONSTRAINT RATING\_CK CHECK(RATING IN('G','PG','R','NC17','NR')),**

**CATEGORY VARCHAR2(20) CONSTRAINT CATEG\_CK CHECK (CATEGORY IN ('DRAMA',**

**'COMEDY','ACTION','CHILD','SCIFI','DOCUMENTARY')),**

**RELEASE\_DATE DATE);**

**Table TITLE created.**

**OUTPUT:**

**Name Null Type**

**------------ -------- -------------**

**TITLE\_ID NOT NULL NUMBER(10)**

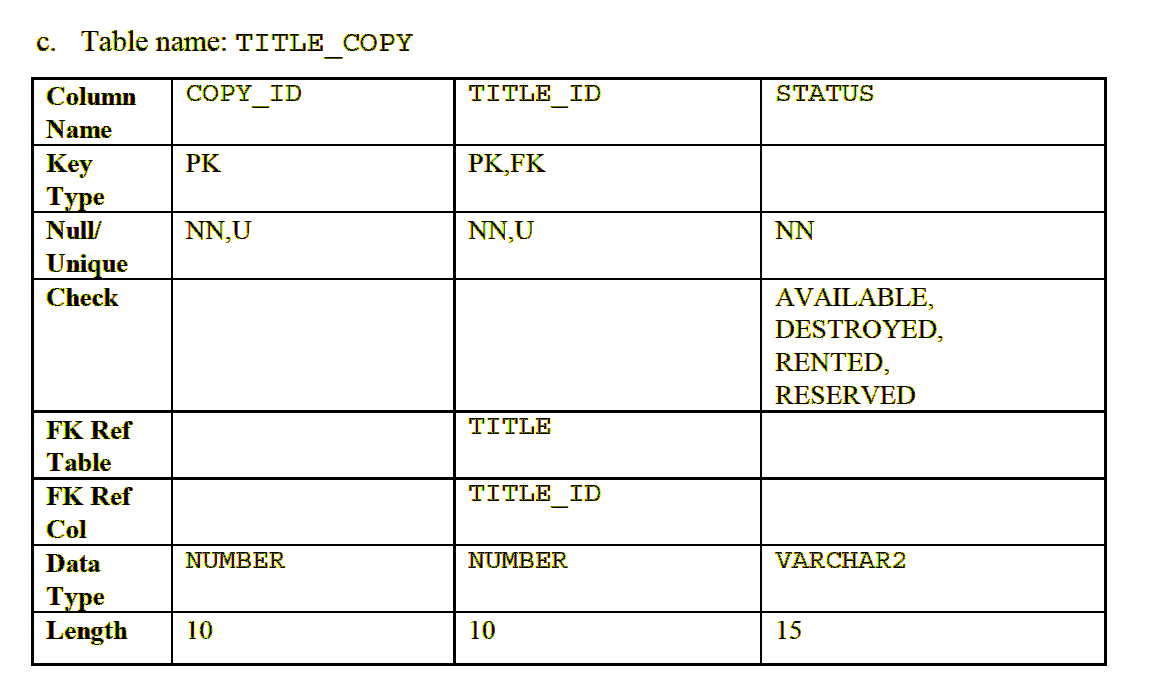
**TITLE NOT NULL VARCHAR2(60)**

**DESCRIPTION NOT NULL VARCHAR2(100)**

**RATING VARCHAR2(4)**

**CATEGORY VARCHAR2(20)**

**RELEASE\_DATE DATE**



**ANSWER:**

**CREATE TABLE TITLE\_COPY (**

**COPY\_ID NUMBER(10) NOT NULL,**

**TITLE\_ID NUMBER(10) NOT NULL UNIQUE CONSTRAINT TITLE\_ID\_FK REFERENCES TITLE(TITLE\_ID),**

**STATUS VARCHAR2(15) NOT NULL CONSTRAINT STATUS\_CK CHECK(STATUS IN ('AVAILABLE', 'DESTROYED', 'RENTED', 'RESERVED')),**

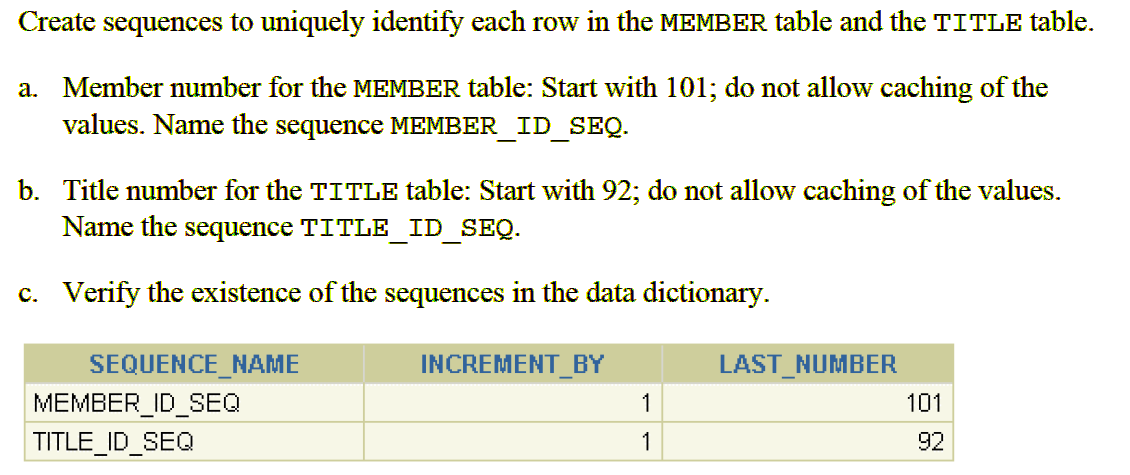
**PRIMARY KEY (COPY\_ID, TITLE\_ID)**

**);**

**Table TITLE\_COPY created.**

**OUTPUT:**

1. **Name Null Type**
2. **-------- -------- ------------**
3. **COPY\_ID NOT NULL NUMBER(10)**
4. **TITLE\_ID NOT NULL NUMBER(10)**
5. **STATUS NOT NULL VARCHAR2(15)**
6. (5 marks)



ANSWER:

A.

**CREATE SEQUENCE MEMBER\_ID\_SEQ**

**INCREMENT BY 1**

**START WITH 101**

**MAXVALUE 9999**

**NOCACHE**

**NOCYCLE;B.**

B.

**CREATE SEQUENCE TITLE\_ID\_SEQ**

**INCREMENT BY 1**

**START WITH 92**

**MAXVALUE 9999**

**NOCACHE**

**NOCYCLE;;**

C.

**SELECT SEQUENCE\_NAME, INCREMENT\_BY, LAST\_NUMBER**

**FROM user\_sequences**

**WHERE sequence\_name IN ('MEMBER\_ID\_SEQ', 'TITLE\_ID\_SEQ');**

**OUTPUT:**

**SEQUENCE\_NAME INCREMENT\_BY LAST\_NUMBER**

**--------------------------------- ------------------------------ --------------------------**

**MEMBER\_ID\_SEQ 1 101**

**TITLE\_ID\_SEQ 1 92**