IZZATI AISYAH BINTI AHMAD
A189633 1MM2

TT1964 DATABASE

TUTORIAL 3

Structured Query Language (SQL)

Instruction: Please answer all of the following questions BEFORE your tutorial session. Then you can have some discussion regarding the questions during your tutorial session. (You can refer flipped material Normalization *http://myftsm.wixsite.com/database-flipped*)

Section A

Creating Table

FORMAT ASAS QUERY :
SELECT _____ (attr.)
FROM _____ (nama table)
WHERE _____ (condition)

1. Write the SQL statement to create this 2 table below with all this condition :

   STUDENT(matricNo,studentName,studentAdd,facNo)

   FACULTY(facNo,facName,facAdd)

   Condition:

   - ✓ assign the matric number & the faculty number as a primary key
   - ✓ assign the faculty number as foreign key for table student
   - ✓ the matric number should be not more than 7 character

```
CREATE TABLE STUDENT (
matricNo varchar(7) NOT NULL ,
studentName varchar(30) NOT NULL,
studentAdd varchar(100) NOT NULL,
facNo integer NOT NULL,
PRIMARY KEY (matricNo),
FOREIGN KEY (facNo) REFERENCES FACULTY (facNo) ) ;


CREATE TABLE FACULTY (
facNo integer NOT NULL ,
facName varchar(30) NOT NULL,
facAdd varchar(100) NOT NULL,
PRIMARY KEY (facNo) ) ;
```

2. Based on this following tables below answer all the questions:

CUSTOMER

cust_no	**cut_name**	**amount**	**branch_name**
L-170	Ahmad	3000	Bangi
L-230	Siti	4000	Cyberjaya

a) Create one table name as above (use suitable datatype)

```
create table CUSTOMER (cust_no varchar (5), cut_name varchar(15) , amount
integer, branch_name varchar(15), PRIMARY KEY (cust_no) )
```

b) Insert all the data above.

```
insert into CUSTOMER values ('L-170' , 'Ahmad' , 3000, 'Bangi' );

insert into CUSTOMER values ('L-230' , 'Siti' , 4000, 'Cyberjaya' );
```

c) The attribute customer has a wrong spelling name when first time you creating the table. Change the attribute name with a right spelling.

```
ALTER TABLE CUSTOMER
RENAME COLUMN cut_name TO cust_name;
```

d) Ahmad has change his branch name to “Cheras”, update the data above.

```
update CUSTOMER
 set branch_name = 'Cheras'
 where cust_name = 'Ahmad' ;
```

e) Siti is no longer customer for this company, update the data above.

```
delete from CUSTOMER
where cust_name = 'Siti' ;
```

Section B

Relational Operations

1. Based on this following tables below answer all the questions:

   Employee (empNo, fName, lName, address, DOB, sex, position, salary,deptNo)

   Department (deptNo, deptName, mgrEmpNo)

   Project (projNo, projName, deptNo)

   WorksOn (empNo, projNo, dateWorked, hoursWorked)

a) List all employees in alphabetical order of surname, and then first name.

```
SELECT fName, lName
FROM Employee
ORDER BY fName, lName, ASC ;
```

b) List the names and addresses of all employees who are Managers.

```
SELECT fName, lName, address
FROM Employee
WHERE position = 'Managers' ;
```

data adalah case sensitive,, yang lain case insensitive (where, select, nama attribute)
" or "" bergantung pada software

c) List the employees' name and salary which their salary is more than 10000.

```
SELECT fName, lName, salary
FROM Employee
WHERE salary>10000 ;
```

d) What is the average salary for the company?

```
SELECT AVG(salary) as companyAVG
FROM Employee ;
```

e) Produce a list of the names and addresses of all employees who work for the IT *department.* kita nak capai data dari 2 table

```
SELECT fName, lName, address
FROM Employee
WHERE deptNo =
        ( SELECT deptNo
          FROM Department
          WHERE deptName = 'IT' ) ;
```

```
SELECT fName, lName, address
FROM Employee e, Department d
WHERE e.deptNo = d.deptNo
AND deptName = 'IT' ;
```

f) Find out how many employees work at IT *department*

```
SELECT empNo
COUNT (empNo) as totalEmp
FROM Empoyee ;
```

```
SELECT COUNT (e.empNo) AS employeeNum
FROM Employee e, Department d
WHERE e.deptNo = d.deptNo
AND deptName = 'IT' ;
```

g) What is the total salary for every department in this company? Create an appropriate heading for the columns of the results table.

```
SELECT deptNo
  COUNT(empNp) AS myCount,
  SUM(salary) AS mySum
FROM Employee
GROUP BY deptNo
ORDER BY deptNo;
```

```
SELECT sum(salary) AS totalSalary, deptName
FROM Employee e, Department d
WHERE e.deptNo = d.deptNp
GROUP by deptName;
```

h) What is the total salary for every department in the company starting from alphabet A?

```
SELECT deptNo
  COUNT(empNp) AS myCount,
  SUM(salary) AS mySum
FROM Employee
GROUP BY deptNo
ORDER BY deptNo;
HAVING deptName LIKE 'A%' ;
```

```
SELECT SUM(salary) AS totalSalary, e.deptName
FROM Employee e, Department d
WHERE e.deptNo = d.deptNo
GROUP BY deptName
HAVING deptName like '%A' ;
```

i) Produce a report of the hours worked by each employee, arranged in order of department number and within department, alphabetically by employee surname.

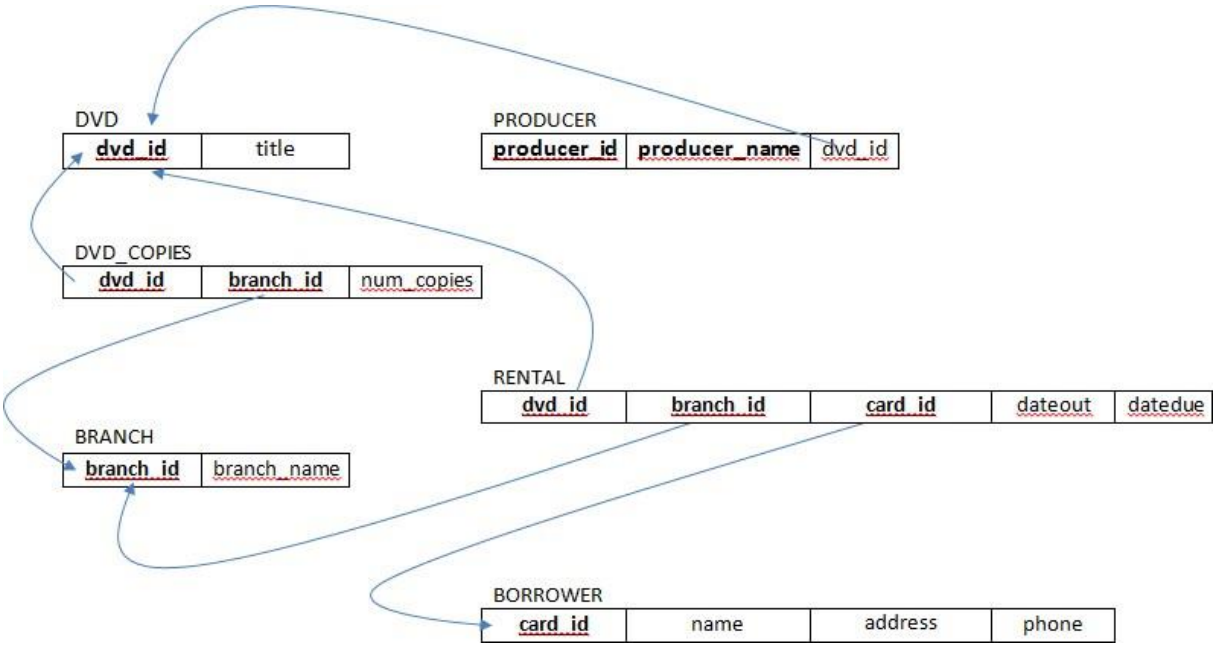
j) For each project on which more than two employees worked, list the project number, project name, and the number of employees who work on that project.

```
SELECT COUNT(empNo) as empCount, p.projNo, p.projName
FROM Project p, WorksOn w
Where p.projNo = w.projNo
AND count (empNo) > 2;
```

k) List the numbers of employees in each department for those departments with more than 10 employees. Create an appropriate heading for the columns of the results table.

```
SELECT count (empNo) as emtTotal, deptName
FROM Employee e, Department
WHERE e.deptNo = d.deptNo
AND count(empNo)>10 ;
```

Nested Queries



a) Retrieve the names of all producers of DVD title is “Spiderman2”

```
SELECT producer_name
FROM DVD d, Producer p
WHERE d.dvd_id = p.dvd_id
AND title = 'Spiderman2' ;
```

```
SELECT producer_name
FROM producer
WHERE IN (SELECT dvd_id
          FROM DVD
          WHERE title = 'Spiderman2') ;
```

b) How many copies of the DVD titled “Rio2” are owned by the DVD house branch whose name is "SPEEDY"?

```
SELECT DVD_COPIES.num_copies
FROM DVD,DVD_COPIES,BRANCH
WHERE DVD.dvd_id = DVD_COPIES.dvd_id
AND BRANCH.branch_id = DVD_COPIES.branch_id
AND TITLE = ‘Rio2’
AND BRANCH= ‘SPEEDY’;
```

c) Retrieve the names of all borrowers who still borrowing the DVD.

```
SELECT name
FROM Borrower
WHERE card_id IN
(SELECT card_id  FROM Rental);
```

d) For each DVD house branch, retrieve the branch name and the total number of DVDs had rented at that branch.

```
SELECT num_copies, branch_name
FROM dvd_copies
WHERE brand_id IN
(SELECT branch_id FROM BRANCH);
```

SAMPLE PROJECT : PROBLEM SOLVING

CHOMEY MINI MART

Based on the previous NORMALIZATION topic, write the SQL statement (Query) to generate nimum 10 **valuable** reports that consists data definition and data manipulation statement for this case study.

NOTES :