Department of Computer Engineering
University of Peradeniya
CO226 – Database Systems

Tutorial 01 – Answers

Question 1
Q1.1

Data definition language:

This is used to define the database conceptual schema. In most DBMSs, the DDL also defines user views and, sometimes, storage structures.
Used to build and modify the structure of your tables and other objects in the database. When you execute a DDL statement, it takes effect immediately. The create table statement does exactly that

Data manipulation language:

A data manipulation language (DML) is used for specifying database retrievals and updates. DMLs can be high level (set-oriented, nonprocedural) or low level (record oriented, procedural). A high-level DML can be embedded in a host programming language, or it can be used as a standalone language; in the latter case it is often called a query language.

DML is Data Manipulation Language: it is used to manipulate data itself. For example, with SQL, it would be instructions such as insert, update, delete.

Primary key:

A primary key is a field in a table which is unique and enables you to identify every record in that table.

Foreign key:

In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In other words, a foreign key is a column or a combination of columns that is used to establish and enforce a link between two tables.

Referential integrity:

Referential integrity is a concept for ensuring that relationships between database tables remain consistent. In other words, references to data must be valid. A relationship between two database tables, called a referenced table and a referencing table, is created by using a foreign key.

Correlated sub query:

In a SQL database query, a correlated subquery (also known as a synchronized subquery) is a subquery (a query nested inside another query) that uses values from the outer query. The subquery is evaluated once for each row processed by the outer query.

Natural join:

A natural join is a type of equi-join where the join predicate arises implicitly by comparing all columns in both tables that have the same column-names in the joined tables. The resulting joined table contains only one column for each pair of equally named columns.

View:

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

Trigger:

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server. DML triggersexecute when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

Q 1.2 Explain the differences in ON UPDATE RESTRICT/CASCADE/SET NULL clauses.

CASCADE

Whenever rows in the master (referenced) table are deleted (or updated), the respective rows of the child (referencing) table with a matching foreign key column will be deleted (or updated) as well. This is called a cascade delete (or update).

RESTRICT

A value cannot be updated or deleted when a row exists in a referencing or child table that references the value in the referenced table.

Similarly, a row cannot be deleted as long as there is a reference to it from a referencing or child table.

SET NULL

The value of the affected referencing attributes is changed to NULL for SET NULL.

Q 1.3 What happens if the ON DELETE CASCADE clause is set?
On delete: delete all corresponding records

Question 2

Consider a MySQL table containing drink information.

DRINKS (ID, Drink_Name, Cost, Color, Ice, Calories)

Perform the following operations on the table DRINKS.

1. Display the average calorie amount of a drink as Avg_Cal.
```
select (sum(calories)/8) as avg_cal from drinks;
```
2. Display the table according to the alphabetical order of the names and the increasing order of the prices.
```
select * from drinks order by drink_name,cost;
```
3. Display only the information from third row to sixth row.
```
select * from drinks limit 2,4;
```
4. Display the names, calories and cost of drinks that contain no more than 30 calories and cost more than Rs. 3.00.
```
select drink_name,calories,cost from drinks where calories<=30 and cost>3.0;
```
5. Display the name and cost of drinks that start with the letter 'B' or that cost less than Rs. 3.00.
```
select drink_name,cost from drinks where drink_name like 'B%' or cost<3.0;
```
6. Display the first two letters of each drink as First_Two_Letters.
```
select left(drink_name,2) as first_two_letters from drinks;
```
7. A person wants to buy one bottle from each drink. Display the total cost for the bottles.
```
select sum(cost) from drinks;
```
8. Rename the table to 'DRINK_INFO' and then change the column Drink_Name to Dname.
```
alter table drinks rename to drink_info;
alter table drink_info change column drink_name dname varchar(20) not null;
```

9. Change drink color values from ‘yellow’ to ‘gold’.
```
update drink_info set color='gold' where color='yellow';
```

10. Make all the drinks that cost Rs. 2.50 to be Rs. 3.50, and make all the drinks that cost Rs. 3.50 to be Rs. 4.50.
```
Update drink_info set cost=4.50 where cost=3.50 AND set cost=3.50
where cost=2.50
```

Question 3

Consider the following database schema.

CLIENTS(<u>CID</u>, Cname)
BRANCHES(<u>BID</u>, Bdesc, Bloc, Cid)
SERVICES(<u>SID</u>**,** Sname, Sfee)
BRANCHES_SERVICES(<u>BID,SID</u>)

Definitions for attributes are as follows.

CID,Cid : Client ID, **Cname**: Client name
BID: Branch id, **Bdesc**: Branch description, **Bloc**: Branch location,
SID: Service ID, **Sname**: Service name, **Sfee**: Service fee

Perform the following queries using both joins and subqueries. You may use the subqueries as specified.

1. Joins or Subqueries within WHERE or HAVING Clause or both.

a) List all branch offices belonging to the client, “Rabbit Foods”.
```
select bdesc from branches as b, client as c where b.cid=c.cid and
cname='Rabbit Foods';
```

b) List all services offered by the client “SED Agency”.
```
select s.sname from client as c,branches as b,branches_services as
bs, services as s where b.cid=c.cid and bs.bid=b.bid and bs.sid=s.sid
and cname ='SED Agency';
```

c) List all clients having exactly two branch offices.
```
select c.cname from client as c, branches as b where b.cid=c.cid
group by c.cid having count(b.bid)=2;
```

d) Find all clients charging a service fee that is the maximum service fee.
```
select distinct cname from client as c, branches as b,
branches_services as bs, services as s where b.cid=c.cid and
bs.bid=b.bid and bs.sid=s.sid and sfee=(select max(sfee) from
services);
```

2. Joins or Subqueries with Logical and Comparison Operators or both.

a) Find all branches offering more than 50% of all available services.
```
select b.bdesc from branches as b, branches_services as bs, services
as s where bs.bid=b.bid and bs.sid=s.sid group by b.bid having
count(s.sid)>(select count(*) from services)/2;
```

b) Find all clients which are offering all available services across their branch offices.
```
select b.bdesc from branches as b,branches_services as bs, services
as s where bs.bid=b.bid and bs.sid=s.sid group by b.bid having
count(s.sid)=(select count(*) from services);
```

3. Joins or Subqueries with IN Membership Test or both.

a) List all services offered by N Region HO branch office.
```
select s.sname from branches as b, branches_services as bs, services
as s where bs.bid=b.bid and bs.sid=s.sid and b.bdesc='N Region HO';
```

b) List all branches offering the Accounting service.
```
select b.bdesc from branches as b, branches_services as bs, services
as s where bs.bid=b.bid and bs.sid=s.sid and s.sname='Accounting';
```

c) List all branches with their customer name offering the accounting service.
```
select b.bdesc,c.cname from client as c, branches as b,
branches_services as bs, services as s where b.cid=c.cid and
bs.bid=b.bid and bs.sid=s.sid and s.sname='Accounting';
```

d) List all clients offering the Accounting services.
```
select distinct c.cname from client as c, branches as b,
branches_services as bs, services as s where b.cid=c.cid and
bs.bid=b.bid and bs.sid=s.sid and s.sname='Accounting';
```

e) List all branches that do not offer the Accounting service.
```
select bdesc from branches where bdesc not in (select b.bdesc from
branches as b, branches_services as bs, services as s where
bs.bid=b.bid and bs.sid=s.sid and s.sname='Accounting');
```

4. Joins or Subqueries with the EXISTS operator or both

a) Check whether there are clients whose branches offering 5 or more services.
```
select cname from client where exists (select bdesc from branches as
b, branches_services as bs where bs.bid=b.bid group by b.bid having
count(bs.sid)>=5);
```

5. Joins or Subqueries in the FROM Clause or both

a) List average number of services offered by each branch.
```
select avg(temp.total) as Average_Branches from (select count(bs.sid)
as total from branches as b, branches_services as bs where
bs.bid=b.bid group by b.bid) as temp;
```

b) List all branches offering services which is above the average number of services

```
select bdesc from branches as b, branches_services as bs where bs.bid=b.bid group by b.bid having count(bs.sid)> (select avg(temp.total) as Average_Branches from (select count(bs.sid) as total from branches as b, branches_services as bs where bs.bid=b.bid group by b.bid) as temp);
```

6. Both Joins and Subqueries

a) List all clients, those do not have any branch, using a Left join and a subquery

```
select cname from client as c left join branches as b on b.cid=c.cid where bid is null;
```

7. Subqueries and Other DML Statements (UPDATE and DELETE)

a) All branches located in California have decided to offer the security service instead of the administration service. Implement this change.

```
update branches_services set sid=(select sid from services where sname='security') where sid=(select sid from services
where sname='administration') and bid in (select bid from branches where bloc='CA');
```

b) Find out the services that are used by three or more branch offices and then increase the fee for those services by 25 percent.

```
update services set sfee=sfee+(sfee*0.25) where sid in (select sid from branches_services group by sid having count(bid)>3);
```

c) Delete all branches using the Recruitment service.

```
delete from branches where bid in (select bid from services as s, branches_services as bs where s.sid=bs.sid and sname='recruitment');
```

d) Delete all clients any of whose branch offices generate service fee revenues of $500 or less.

```
delete from client where cid in (select distinct cid from branches as b, services as s, branches_services as bs where
b.bid=bs.bid and s.sid=bs.sid group by bs.bid having sum(sfee<=500));
```