

# 1.

## a.) *base table; view.*

- Base tables store raw data and not derived data.
- A view is a query result from base table(s)/view(s).
- A view may join multiple tables or views together and may contain derived columns

## b.) *catalog; schema.*

- Schema contains descriptions of objects such as base tables, views, constraints, etc.
- The catalog is a set of schemas

## c.) *where-clause; having-clause.*

- Where-clause is for specifying conditions on selecting *a single row* in the SELECT command, while having-clause is for specifying conditions for selecting *groups of rows* under the Group-By clause of the SELECT command.
- Example: select only customers with an outstanding balance (“where balance >0”); Group customer records by state where there are more than 5 customers (“having”)

## **2. Explain how a view can be used to reinforce data security.**

- Tables and columns that are not included in a view will not be exposed to the user or the programmer of the view, minimizing the risk of unauthorized access or data corruption.
- Restricting access to a view with GRANT and REVOKE statements (security statements) adds a further layer of security.

**3.**

**a) Define the tables for Student, Course, Tutorial, and IsRegistered.**

```
CREATE TABLE Student
(StudentID  NUMBER(8)    NOT NULL,
StudentName CHAR(25),
CONSTRAINT StudentPK PRIMARY KEY (StudentID));
```

```
CREATE TABLE Course
(CourseID  CHAR(8)    NOT NULL,
CourseName CHAR(15),
CONSTRAINT CoursePK PRIMARY KEY (CourseID));
```

```
CREATE TABLE Tutorial
(TutorialID  NUMBER(8)    NOT NULL,
CourseID     CHAR(8),
CONSTRAINT TutorialPK PRIMARY KEY (TutorialID),
CONSTRAINT TutorialFK FOREIGN KEY (CourseID)
REFERENCES Course (CourseID));
```

```
CREATE TABLE IsRegistered
(StudentID  NUMBER(8)    NOT NULL,
TutorialID NUMBER(8)    NOT NULL,
Semester   CHAR(7),
CONSTRAINT IsRegisteredPK PRIMARY KEY (StudentID,
TutorialID),
CONSTRAINT IsRegisteredFK1 FOREIGN KEY(StudentID)
REFERENCES Student(StudentID),
CONSTRAINT IsRegisteredFK FOREIGN KEY (TutorialID)
REFERENCES Tutorial(TutorialID));
```

**b) List the course ID and course name for all courses with CourseIDs starting with 'ISM'.**

```
SELECT CourseID, CourseName
FROM Course
WHERE CourseID LIKE 'ISM%';
```

**c) Define a student list as a view for all students registered in semester I-98.**

```
CREATE VIEW StudentI98V AS
  SELECT DISTINCT StudentID
  FROM IsRegistered
  WHERE Semester = 'I-98';
```

**d) Find which lecturers are qualified to teach ISM3113 before 01-JAN-2000.**

```
SELECT LecturerID
FROM IsQualified
WHERE CourseID = 'ISM3113'
AND DateQualified < '01-JAN-2000';
```

**e) Find any lecturers who can teach both ISM3113 and ISM3114.**

Will the following SQL work?

```
SELECT LecturerID
FROM IsQualified
WHERE CourseID = 'ISM3113'
AND CourseID = 'ISM3114';
```

For example, if IsQualified table has the following data:

<u>LecturerID</u>	<u>CourseID</u>	DateQualified
1001	ISM3113	01-JAN-2000
1001	ISM3114	03-JAN-2000
1002	ISM3113	11-JAN-2000
1003	ISM3113	21-JAN-2000
1003	ISM3114	21-JAN-2000
1004	ISM3114	23-FEB-2001

1005	ABS2001	23-FEB-2001
------	---------	-------------

The above SQL will return no results, since none of the rows satisfies the condition in the “WHERE” clause.

The information that a lecturer can teach 2 tutorials is stored in 2 rows. Hence we cannot use a single “WHERE” clause to specify the condition.

Solution 1:

```
SELECT LecturerID, COUNT(*)  
FROM IsQualified  
WHERE CourseID = 'ISM3113'  
       OR CourseID = 'ISM3114'  
GROUP BY LecturerID  
HAVING COUNT(*) = 2;
```

The query result will be all faculty IDs who can teach both ISM3113 and ISM3114:

LecturerID	COUNT(*)
1001	2
1003	2

Solution 2:

```
SELECT LecturerID
FROM (SELECT LecturerID, COUNT(*) AS N
      FROM IsQualified
      WHERE CourseID IN ('ISM3113','ISM3114')
      GROUP BY LecturerID)
WHERE N=2;
```

The subquery above (in red) is similar to Solution 1. The output of the subquery is:

LecturerID	N
1001	2
1002	1
1003	2
1004	1

The outer query result will be:

LecturerID
1001
1003

Solution 3:

```
SELECT LecturerID
FROM IsQualified
WHERE CourseID = 'ISM3113'
AND LecturerID IN
  (SELECT LecturerID
   FROM IsQualified
   WHERE CourseID = 'ISM3114');
```

The subquery (in red) result will be a subset of lecturers who are qualified for ISM3114:

LecturerID
1001
1003
1004

The outer query result will be lecturers who are qualified for both ISM3113 and ISM3114:

LecturerID
1001
1003



**f) Find out how many students are enrolled in Tutorial 2714 during semester I-98.**

```
SELECT COUNT (StudentID)
FROM IsRegistered
WHERE TutorialID = 2714
AND Semester = 'I-98';
```

**g) Find out how many students are enrolled during semester I-98.**

```
SELECT COUNT (DISTINCT (StudentID))
FROM IsRegistered
WHERE Semester = 'I-98';
```

**h) Insert a student ‘Joe Sun’ with student ID 98765 into the database.**

```
INSERT INTO Student VALUES (98765, ‘Joe Sun’);
```