

# DBS Assignment I

## 1. With necessary examples, explain any four major disadvantages of keeping organizational information in a file-processing system.

- **Data redundancy and inconsistency** - Since different programmers create files and application programs over a long period, the files are likely to have different formats and the programs may be written in several programming languages. Moreover, the same information may be duplicated in several places (multiple files).

For eg, the address and telephone number of particular customer may appear in a file that consists of savings-account records and in a file that consists of checking-account records.

- **Difficulty in accessing data** - Conventional file-processing environments do not allow the needed data to be retrieved in a convenient and efficient manner. More Responsive data-retrieval systems are required for general use.

For eg, various file formats need different parsing handlers.

- **Integrity problems** - The data stored in the database must satisfy certain types of consistency constraints.

For eg, the balance of a bank account may never fall below a prescribed amount (say, 250 Imperial credits). Developers enforce these constraints in the system by adding appropriate code in the various application programs. However, it is difficult to change the programs to enforce new constraints. The problem is harder when constraints involve several data items from different files.

- **Atomicity problems** - A computer system can fail at times, as any other electrical device. In many applications, it is crucial that, if a failure occurs, the data be restored to the consistent state that existed prior to the failure.

For eg, consider a program to transfer 500 credits from account A to account B. If a system failure occurs during the execution of the program, it is possible that the \$50 was removed from account A but was not credited to account B, resulting in an inconsistent database state. This inconsistent state must be removed. It is difficult to ensure atomicity in a conventional file-processing system.

2. Distinguish between the following with necessary examples:
- Logical level and Physical level
  - Instance and Schema
  - Super Key and Candidate Key
  - Primary Key and Foreign Key

LOGICAL LEVEL	PHYSICAL LEVEL
What data is stored in the database along with all the relationships between it.	Physical representation of the database in the system
Represents entities, attributes, relations, constraints on data, seismic information on data, security, integrity information.	Physical implementation of the database to achieve optimal runtime performance and space optimization.

INSTANCE	SCHEMA
Data stored in the database at a particular moment of time is called an instance.	Design of the database is called schema.
Defines the variable declarations in tables that belong to a particular database.	Is of three types - Physical, logical and view.

SUPER KEY	CANDIDATE KEY
A superkey is a set of one or more attributes that, taken collectively, allow us to identify uniquely an entity in the entity set.	Candidate Keys are super keys for which no proper subset is a super key. In other words candidate keys are minimal super keys.
For example, consider an employee table - Employee ( EmployeeID, FullName, SSN, DeptID )	
Super keys - EmployeeID + FullName	Candidate Keys - EmployeeID, SSN

PRIMARY KEY	FOREIGN KEY
Primary key uniquely identify a record in the table.	Foreign key is a field in the table that is primary key in another table.
Cannot accept null values.	Can accept multiple null values.
Can only have one primary key in a table.	Can have multiple foreign keys in a table.

**3. Consider the following insurance database:**

Person (driver-id#, name, address)

Car (license, model, year)

Accident (report-number, date, location)

Owns (driver-id#, license)

Participated (driver-id#, license, report-number, damage-amount)

**Give an expression in SQL to express each of the following queries:**

**a. Find the details of all the Accidents held in Manipal and where 'Mohan' is involved**

```
SELECT *
FROM Accident
WHERE location = 'Manipal' AND report-number IN
    (SELECT report-number
     FROM Participated
     WHERE driver-id# IN
         (SELECT driver-id#
          FROM Person
          WHERE name IS 'MOHAN'))
);
```

**b. Find the no. of people who has purchased one or more 'Baleno' in 2015**

```
SELECT Count(*)
FROM Person
WHERE driver-id# IN
    (SELECT driver-id#
     FROM Owns
     WHERE license IN
         (SELECT license
          FROM Car
          WHERE model IS 'Baleno' AND year = '2015'))
);
```

**c. Find the highest selling model in the year 2015**

```
SELECT FIRST(model)
FROM Car
WHERE year = '2015'
GROUP BY model
ORDER BY model DESC;
```

**d. Find the date where total damage\_amount of different accidents is maximum.**

```
SELECT FIRST(date)
FROM Accident, Participated
WHERE Participated.report-number = Accident.report-number
GROUP BY report-number
ORDER BY damage-amount DESC;
```

**4. Consider ODI Cricket Database given below.**

Match (MatchID, team1, team2, ground, date, winner)

Player (PlayerID, Lname, Fname, Country)

Batting (MatchID, PlayerID, Order, HOut, NoOfRuns, Mts, NoOfballs, Fours, Sixes)

Bowling (MatchID, PlayerID, NoOfOvers, Maidens, NoOfRuns, NoOfWickets)

**Give an expression in SQL to express each of the following queries:**

- a. List the name of all the players who has done batting in 'Chinnaswamy Stadium' in some ODI**

```
SELECT Fname, Lname
FROM Player
WHERE PlayerID IN
    (SELECT PlayerID
     FROM Batting
     WHERE MatchID IN
         (SELECT MatchID
          FROM Match
          WHERE ground IS 'Chinnaswamy Stadium'))
);
```

- b. Find the Player ID who has scored, more than maximum score obtained by any player in 'Chinnaswamy Stadium', in some ODI.**

```
SELECT PlayerID
FROM Player
WHERE PlayerID IN
    (SELECT PlayerID
     FROM Batting
     WHERE NoOfRuns >
         (SELECT MAX(NoOfRuns)
          FROM Batting
          WHERE MatchID IN
              (SELECT MatchID
               FROM Match
               WHERE ground IS 'Chinnaswamy Stadium'))
    )
);
```

- c. Find the Match ID, played in 'Chinnaswamy Stadium' which has seen one or more centuries.**

```
SELECT MatchID
FROM Match
WHERE ground IS 'Chinnaswamy Stadium' AND
    MatchID IN
        (SELECT MatchID
         FROM Batting
         WHERE NoOfRuns >= 100);
```

**d. Find the name of the players who has scored century in every ground**

```
SELECT Fname, Lname
FROM Player
WHERE PlayerID IN
    (SELECT PlayerID
     FROM Batting
     WHERE NoOfRuns > 99 AND MatchID IN
        (SELECT DISTINCT ground, MatchID
         FROM Match));
```

**5. Give examples to illustrate each of the following constructs:**

**i. from employee as e1, employee as e2 ....**

**ii. > some**

**iii. not exists**

**iv. with**

**using employee database:**

```
Employee (employee-name, street, city)
Works (employee-name, company-name, salary)
Company (company-name, city)
Manages (employee-name, manager-name)
```

i) 

```
SELECT employee-name
FROM Employee as E1, Employee as E2
WHERE E1.city = E2.city AND E1.city = 'Cloud City'
```

ii) 

```
SELECT employee-name
FROM Works
WHERE salary > some (SELECT salary
                     FROM Works
                     WHERE company-name = 'The Empire')
```

iii) 

```
SELECT employee-name
FROM Employee
WHERE NOT EXISTS
    (SELECT employee-name
     FROM Works
     WHERE city = 'Dagobah')
```

iv) 

```
WITH Emp AS (SELECT * FROM Employee)
SELECT *
FROM Emp
WHERE city = 'Tatoonie'
UNION ALL
SELECT *
FROM Emp
WHERE Emp.employee-name IN
    (SELECT employee-name
     FROM Works
     WHERE salary GREATER THAN 9000)
```