6.2 : Data Views

IT2306 - Database Systems

Level I - Semester 2

Detailed Syllabus

1. Characteristics of user views

2. View definition and use

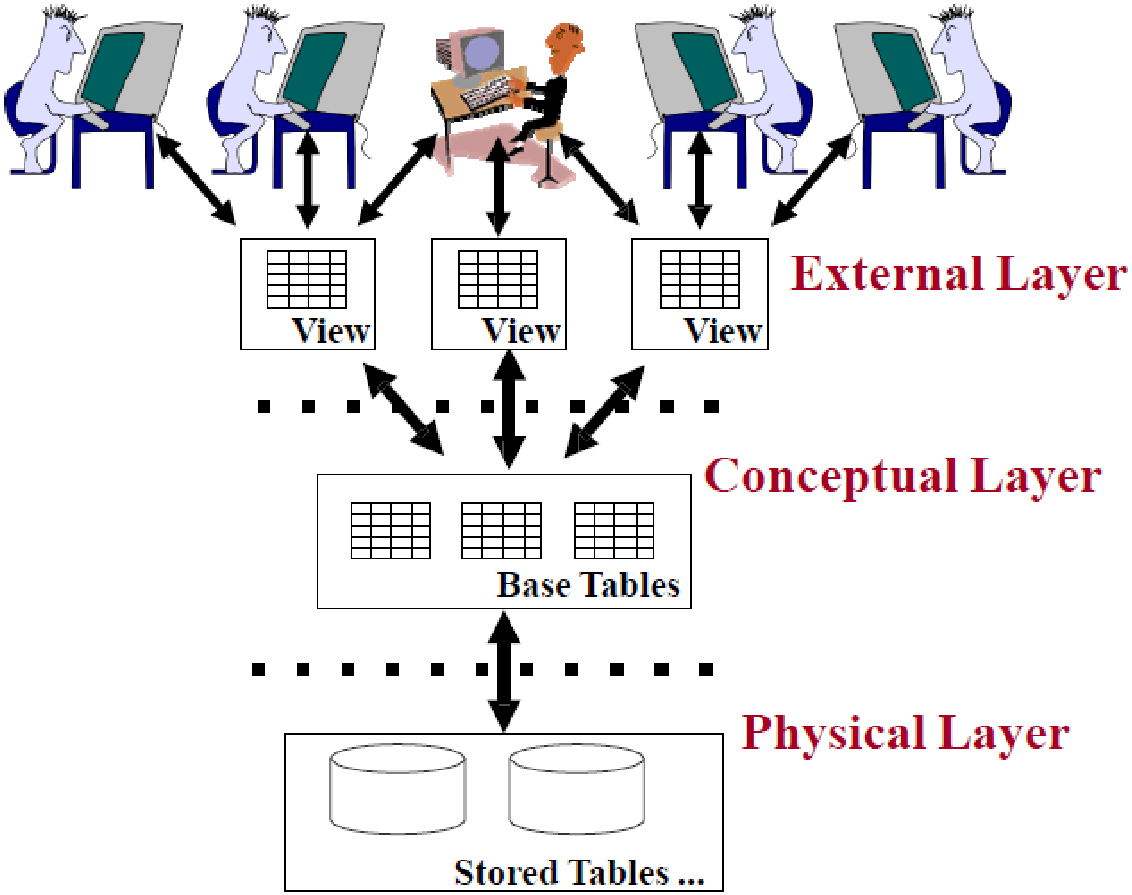
    CREATE VIEW, DROP VIEW, UPDATE VIEW

3. Database Security

    GRANT, REVOKE

1. Characteristics of user views

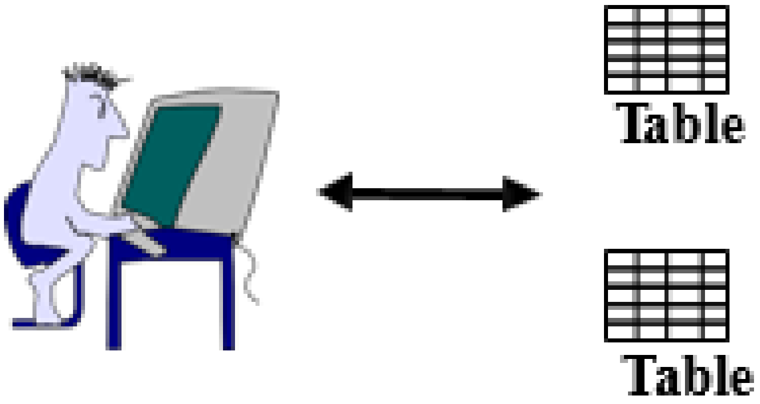
Introduction



1. Characteristics of user views

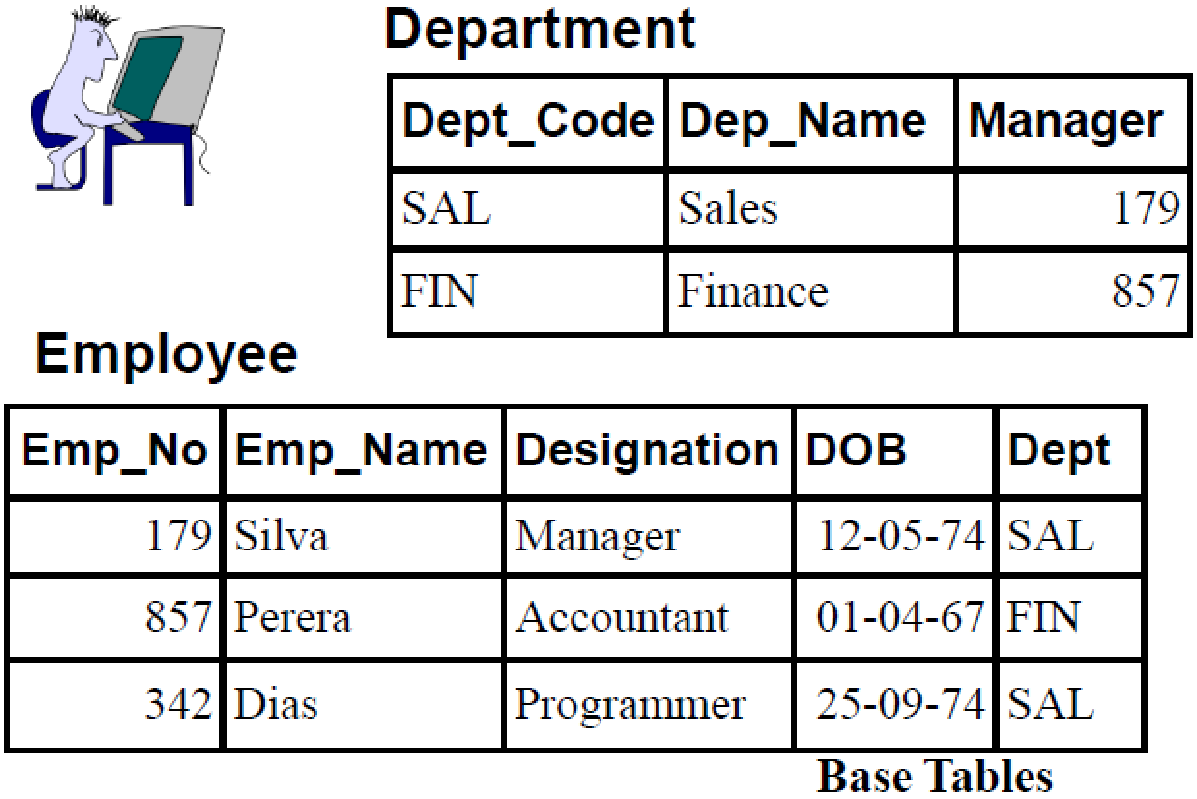
Architecture - Conceptual Layer

- The conceptual model is a logical representation of the entire contents of the database.
- The conceptual model is made up of base tables.
- Base tables are “real” in that they contain physical records.



1. Characteristics of user views

Architecture - Conceptual Layer



Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857

Employee

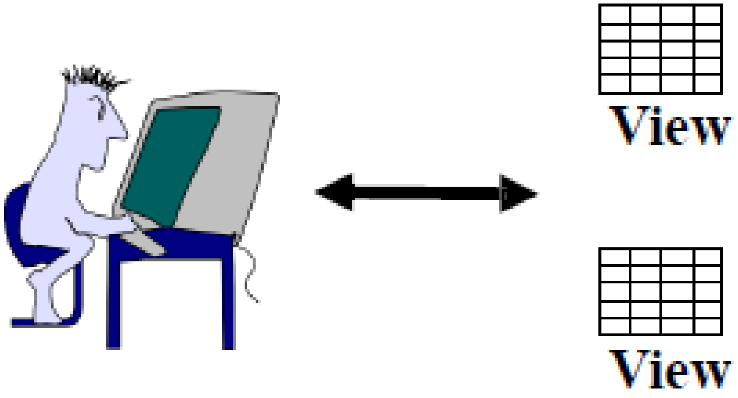
Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

Base Tables

1. Characteristics of user views

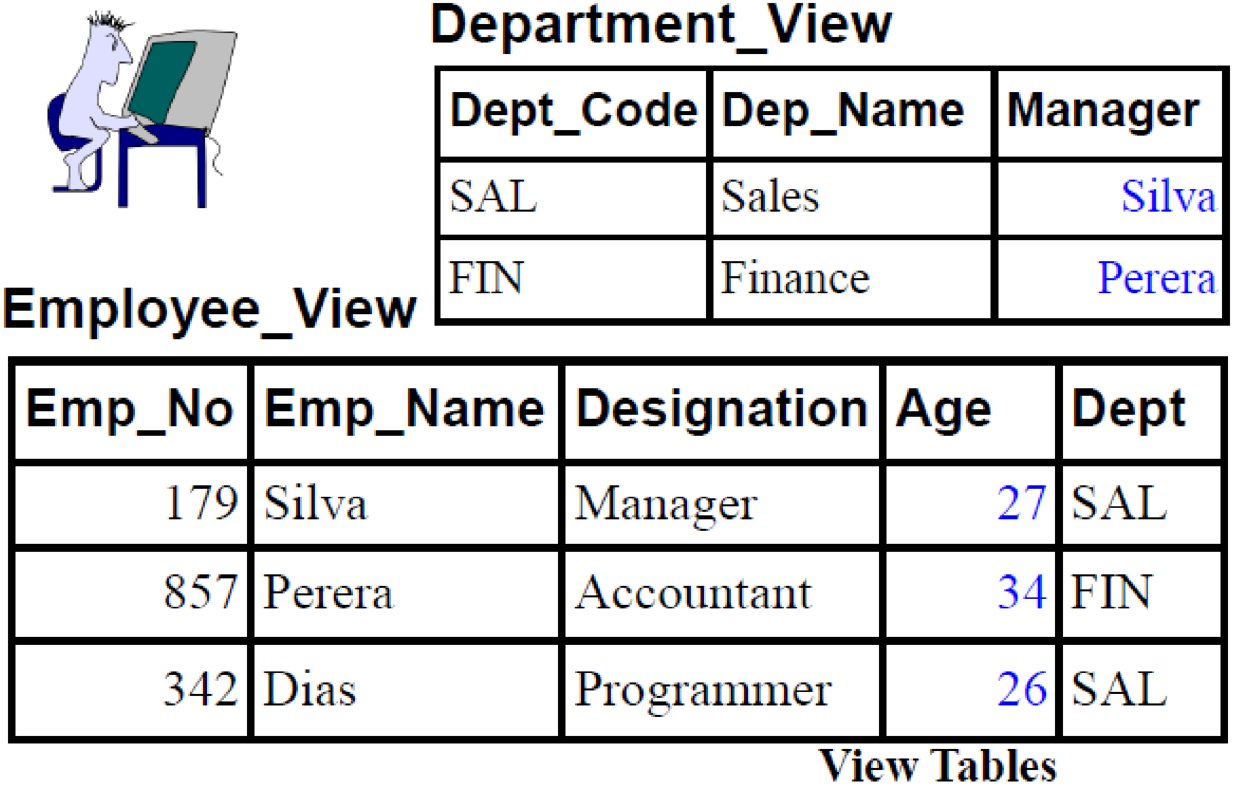
Architecture - External Layer

- The external model represents how data is presented to users.
- The external model is made up of view tables.
- View tables are "virtual"-- they do not exist in physical storage, but appear to a user as if they did.



1. Characteristics of user views

Architecture - External Layer



Department_View

Dept_Code	Dep_Name	Manager
SAL	Sales	Silva
FIN	Finance	Perera

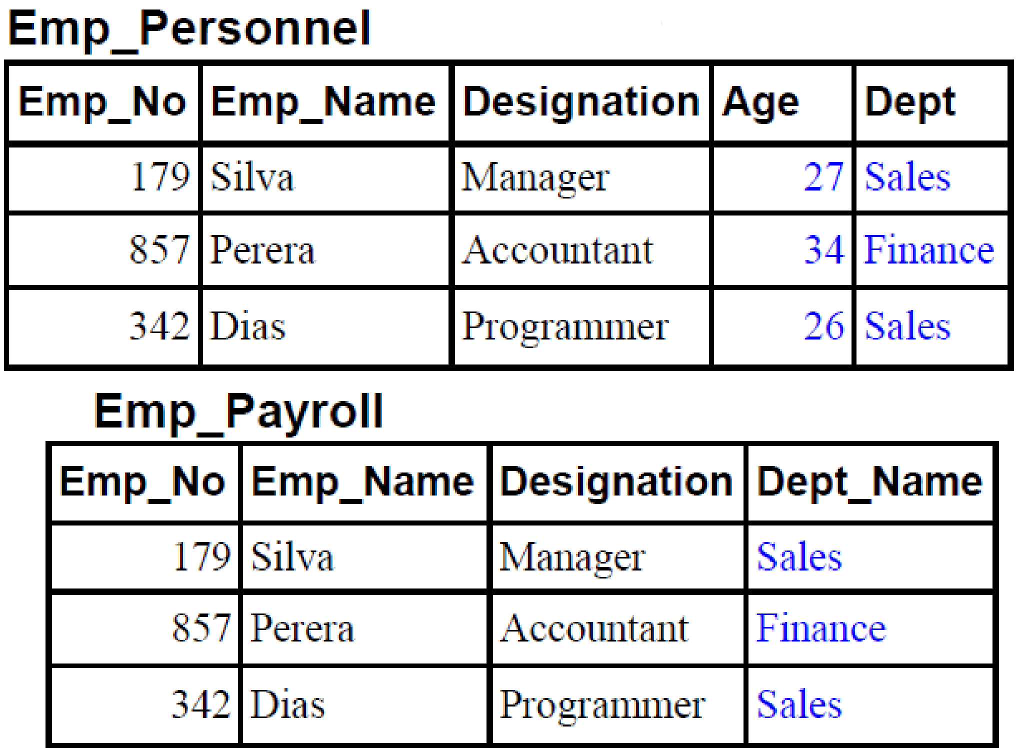
Employee_View

Emp_No	Emp_Name	Designation	Age	Dept
179	Silva	Manager	27	SAL
857	Perera	Accountant	34	FIN
342	Dias	Programmer	26	SAL

View Tables

1. Characteristics of user views

Architecture - External Layer



Emp_Personnel

Emp_No	Emp_Name	Designation	Age	Dept
179	Silva	Manager	27	Sales
857	Perera	Accountant	34	Finance
342	Dias	Programmer	26	Sales

Emp_Payroll

Emp_No	Emp_Name	Designation	Dept_Name
179	Silva	Manager	Sales
857	Perera	Accountant	Finance
342	Dias	Programmer	Sales

1. Characteristics of user views

Architecture - External Layer

Emp_Pay

Emp_No	Emp_Name	Designation	Dept
179	Silva	Manager	SAL
857	Perera	Accountant	FIN
342	Dias	Programmer	SAL

View Tables

Base Tables

Employee

Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

1. Characteristics of user views

Architecture - External Layer

Emp_Personnel

Emp_No	Emp_Name	Designation	Age	Dept
179	Silva	Manager	27	Sales
857	Perera	Accountant	34	Finance
342	Dias	Programmer	26	Sales

View Tables

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857

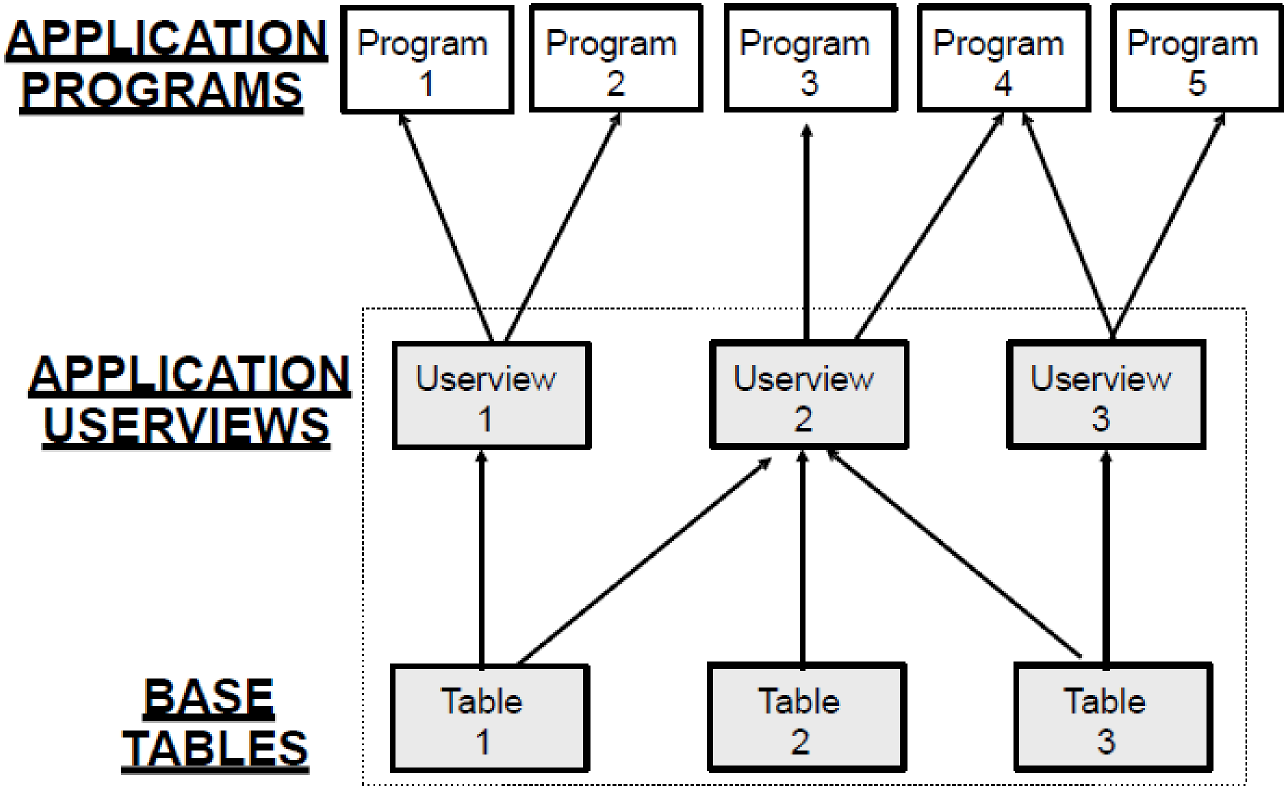
Employee

Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

Base Tables

1. Characteristics of user views

User view Designed for Application



1. Characteristics of user views

What Are User views?

- A view is a virtual table.
- User views
  - Derived or virtual tables that are visible to users.
  - Do not occupies any storage space.
- Base Tables
  - Store actual rows of data.
  - Occupies a particular amount of storage space.

1. Characteristics of user views

- Behave as if it contains actual rows of data, but in fact contains none.
- Rows are derived from base table or tables from which the view is defined.
- Like base tables, views can be queried, updated, inserted into and deleted from, with restrictions.
- Integrity constraints and triggers cannot be defined explicitly for views.
- Usually designed for end users thus providing natural interfaces.

2. View definition and use

Syntax for creating a view :

```
CREATE VIEW view-name
AS query;
```

SQL User Views

- Column names specified must have the same number of columns derived from the query.
- Data definitions for each column are derived from the source table.
- Columns will assumed corresponding column names in the source table.
- Names must be specified for calculated or identical columns.

2. View definition and use

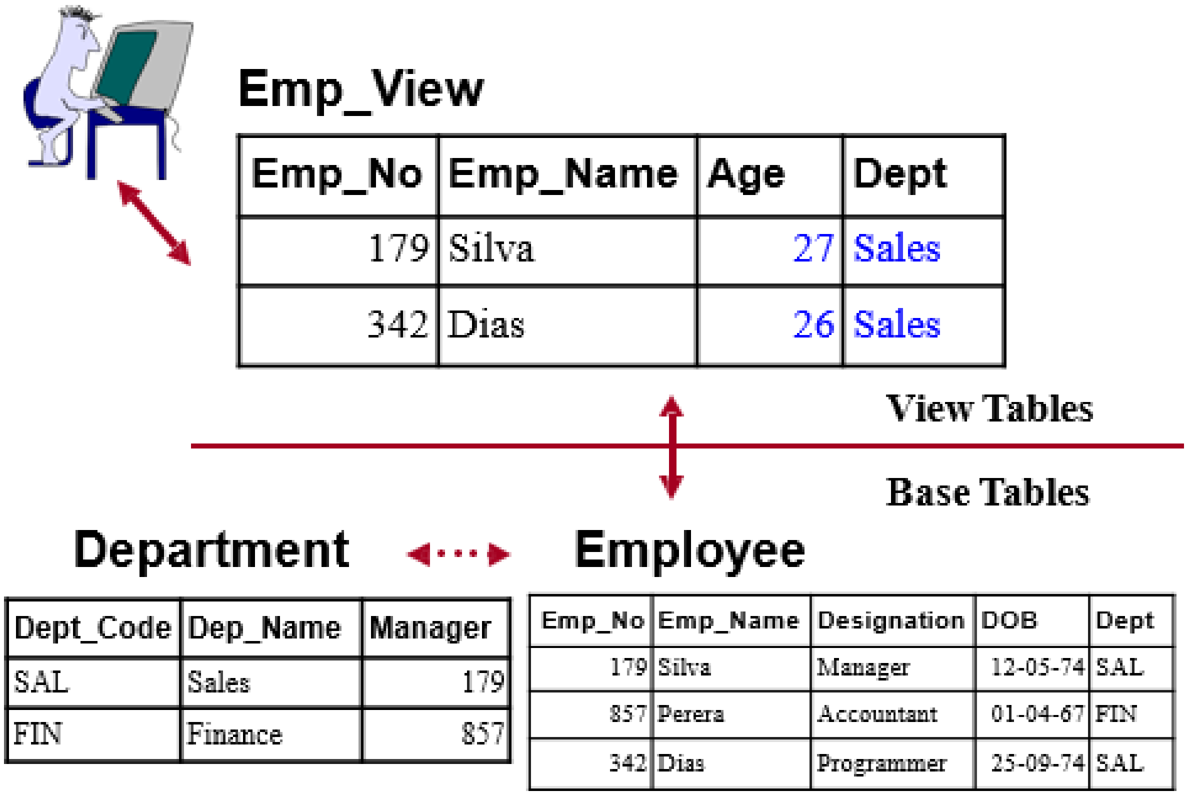
Mechanics of Views

- A view's definition is represented by storing in the data dictionary the text of query that defines the view.

```
CREATE VIEW Emp_View AS
SELECT Emp_No, Emp_Name, Age, Dept_Name
FROM Employee, Department
WHERE Emp.Dept = Dept.Dept_Code and Dept.Dept_Code = ‘SAL’;
```

2. View definition and use

Mechanics of Views



Emp_View

Emp_No	Emp_Name	Age	Dept
179	Silva	27	Sales
342	Dias	26	Sales

View Tables

Base Tables

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857

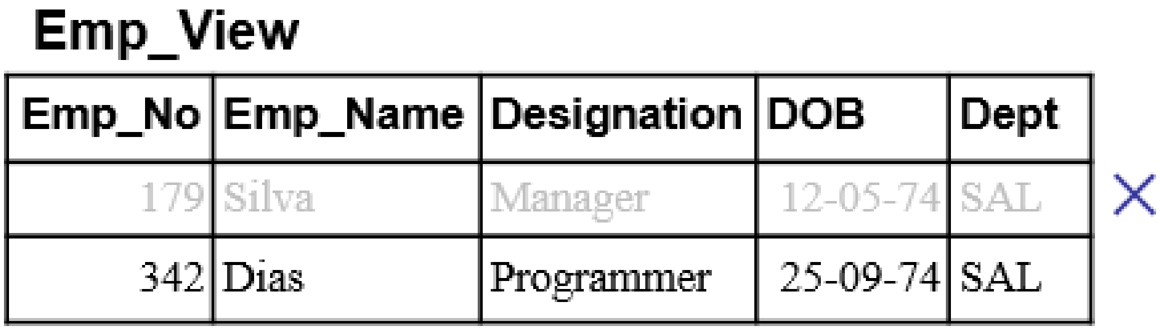
Employee

Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

2. View definition and use

User View Access

```
SELECT *
FROM Emp_View
WHERE Designation='Programmer';
```



Emp_View

Emp_No	Emp_Name	Designation	DOB	Dept	
179	Silva	Manager	12-05-74	SAL	✕
342	Dias	Programmer	25-09-74	SAL	

2. View definition and use

Types of User views

- Selection (Horizontal or Row)

```
CREATE VIEW Sales_Employees AS
SELECT *
FROM Employee
WHERE Dept = ‘SAL’;
```

- Projection (Vertical or Column)

```
CREATE VIEW Emp_Info AS
SELECT Emp_No, Emp_Name, Designation
FROM  Employee;
```

2. View definition and use

Types of User views

- Row/Column

```
CREATE VIEW Sales_Emp_Info AS
SELECT Emp_No, Emp_Name, Designation
FROM Employee
WHERE Dept = ‘SAL’;
```

- Summarized

```
CREATE VIEW Dept_Employees
(Dept_Code,No_Employees) AS
SELECT Dept, COUNT(*)
FROM Employee
GROUP BY Dept;
```

2. View definition and use

Types of User views

- Derive

```
CREATE VIEW Emp_Age_View AS
SELECT *, YEAR(CURRENT)-YEAR(DOB) AS Age
FROM Employee
GROUP BY Dept;

CREATE VIEW Item_Price AS
SELECT *, Dol_Price*90 AS Rs_Price
FROM Items;
```

NOTE: CURRENT gives the system date. YEAR is a built in function which returns a four digit integer that represent the year.

2. View definition and use

Types of User views

- Joined

```
CREATE VIEW Emp_Payroll AS
SELECT Emp_No, Emp_Name, Designation,
Dept_Name
FROM  Employee E, Department D
WHERE E.Dept = D.Dept_Code;
```

2. View definition and use

What are the benefits of having user views?

- Security
  - Protect data from unauthorized access.
  - Each user is given permission to access the database via only a small set of views that contain specific data the user is authorized to see.
- Natural Interface
  - “Personalized” view of database structure, that make sense for the user.
  - Restructure or tailor the way in which tables are seen, so that different users see it from different perspectives, thus allowing more natural views of the same enterprise (e.g. item names)

2. View definition and use

- Query Simplicity
  - Turning multiple table queries to single table queries against views, by drawing data from several tables.
  - It provides flexible and powerful data access capabilities.
  - It also improves productivity of end-user and programmers by:
    - Simplifying database access by presenting the structure of data that is most natural to the user.
    - Simplifying the use of routine and repetitive statements.
    - Building up SELECT statements in several steps.

2. View definition and use

- Insulation from change
  - Data independence - maintain independence among different user views and between each user view and the physical constructs.
  - A view can present a consistent image of the database structure, even if the underlying source tables are restructured.
- Data Integrity
  - DBMS can check data to ensure that it meets specified integrity constraints.

2. View definition and use

User view Design Considerations

- User view design is driven by specific application requirements.
- User may be defined for individual user, or a group of users, of the transaction or application.
- User view may be defined to control and restrict access to specific columns and/or rows in one or more tables.
- User views can be defined to help simplify queries, application development and maintenance.
- User views may be derived from base tables or other user views.

2. View definition and use

Limitations of User views

- Restrictions on views processing
  - SELECT, INSERT, UPDATE and DELETE statements may refer to views, but there are a number of limitations.
  - Update may be possible for ‘simple’ views but not ‘complex’ views.
- Performance
  - DBMS must translate queries against the view to queries against the source tables.
  - These disadvantages means that we cannot indiscriminately define and use views instead of source tables.

2. View definition and use

Remove a User View

- Drop a View

Syntax: DROP VIEW view-name;

E.g. DROP VIEW Emp_Payroll;

- Removes only the definition of the view table.
- Data it used to retrieve is not effected.

2. View definition and use

Update a User View

```
Syntax: UPDATE WORKS_ON
        SET Product_name = ‘ProductY’
        WHERE Emp_Code = 101;
```

- Updating a view defined on a single table without any aggregate functions, can be mapped to an update on the underlying base table.
- Updating a view defined on multiple tables involving joins, can be mapped to an update on the underlying base relations in multiple ways.

2. View definition and use

- A view with a single defining table is updatable if the view attributes contain the primary key of the base relation, as well as all attributes with the NOT NULL constraint that do not have default values specified.
- Views defined on multiple tables using joins are generally not updatable.
- Views defined using grouping and aggregate functions are not updatable.

2. View definition and use

- In SQL, the clause WITH CHECK OPTION must be added at the end of the view definition if a view is to be updated.
- This allows the system to check for view updatability and to plan an execution strategy for view updates.
- It is also possible to define a view table in the FROM clause of an SQL query.
- This is known as an in-line view. In this case, the view is defined within the query itself.

3. Database Security

- Database Security refers to protecting the database from unauthorized or malicious use.
  - Theft of information
  - Unauthorized modification of data
  - Unauthorized destruction of data
- A view is a mean of providing a user with a personalized model of the database.
- It is also a useful way of limiting a user’s access to various portions of the database.
- This simplifies system usage while promoting security.

3. Database Security

Types of View Access

- Read authorization
  - Allows reading, but not modification of data.
- Insert authorization
  - Allows insertion of new data, but no modification of data.
  - Insertion can be for some of the visible attributes and the remaining will take default or NULL values.

3. Database Security

Types of View Access

- Update authorization
  - Allows modification of data but not deletion.
  - Modifications can be for some of the visible attributes.
- Delete authorization
  - Allows deletion of data.

A user may be assigned all, none or a combination of these types of authorization.

3. Database Security

SQL for Data Control

- Security and Access Control can be done using GRANT and REVOKE commands.
- Security and Access Control is based on three central objects:
  - Users
  - Database objects
  - Privileges (select, insert, update, delete, references)

3. Database Security

GRANT Command

- Specify privileges for users on database objects.

Syntax :

```
GRANT <privilege list>
ON  <relation or view>
TO <user list>
```

Example:

```
GRANT   SELECT,  INSERT
ON Employee  TO Silva;
```

```
GRANT   UPDATE(Designation)
ON Employee TO Silva, Dias;
```

3. Database Security

GRANT Command

- The SQL references privilege is granted on specific attributes (as for update).
- This allows a user to create relations that reference an attribute (key) of a relation as foreign key.

```
GRANT REFERENCES(Emp_No)
ON Employee
TO Silva;
```

3. Database Security

REVOKE Command

- Remove privileges from users on database objects.

```
REVOKE <privilege list>
ON <relation or view>
FROM   <user list>
```

Example:

```
REVOKE SELECT
ON Employee
FROM Silva;

REVOKE UPDATE(Designation)
ON Employee
FROM Silva;
```

3. Database Security

SQL-92

- The SQL-92 standard specifies a primitive authorization mechanism for the database schema.
- Only the owner of the schema can carry out any modification to the schema.
- Creating or deleting relations, adding or dropping attributes of relations, adding or dropping indices.