**Practical No. :4**

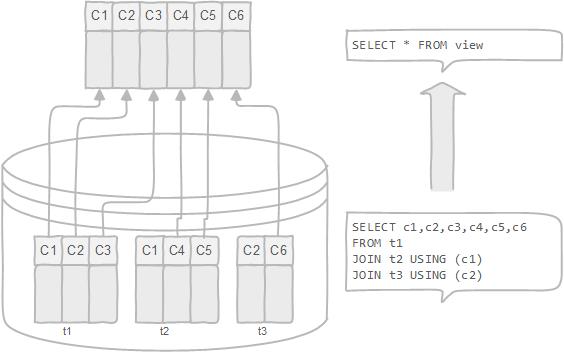
Exercise 5.Transaction processing and Views

***VIEWS in POSTGRESQL***

A view is a database object that is of a stored query. A view can be accessed as a virtual table in PostgreSQL. In other words, a PostgreSQL view is a logical table that represents data of one or more underlying tables through a [SELECT statement](http://www.postgresqltutorial.com/postgresql-select/). Notice that a view does not store data physically except for a [materialized view](http://www.postgresqltutorial.com/postgresql-materialized-views/). A view can represent a subset of a real table, selecting certain columns or certain rows from an ordinary table. A view can even represent joined tables.

A view is named query that provides another way to present data in the database tables. A view is defined based on one or more tables, which are known as base tables. When you create a view, you basically create a query and assign it a name; therefore a view is useful for wrapping a commonly used complex query.

In PostgreSQL, you can create a special view called a materialized view that stores data physically and refreshes the data periodically from the base tables. The materialized views have many advantages in many scenarios such as faster access to data from a remote server, data caching, etc



A view can be very useful in some cases such as: (Advantages)

* A view helps simplify the complexity of a query because you can query a view, which is based on a complex query, using a simple SELECT statement.
* Like a table, you can grant permission to users through a view that contains specific data that the users are authorized to see.
* A view provides a consistent layer even the columns of underlying table changes.
* Structure data in a way that users or classes of users find natural or intuitive.
* Restrict access to the data such that a user can only see limited data instead of complete table.
* Summarize data from various tables, which can be used to generate reports.

**Creating Views:**

PostgreSQL views can be created from a single table, multiple tables, or another view.

### Example

Consider, the COMPANY table is having the following records

id | name | age | address | salary

----+-------+-----+------------+--------

1 | Paul | 32 | California | 20000

2 | Allen | 25 | Texas | 15000

3 | Teddy | 23 | Norway | 20000

4 | Mark | 25 | Rich-Mond | 65000

5 | David | 27 | Texas | 85000

6 | Kim | 22 | South-Hall | 45000

7 | James | 24 | Houston | 10000

Create table company (id int, name varchar(20), age int, address varchar(20), salary int);

CREATE VIEW COMPANY\_VIEW AS

SELECT ID, NAME, AGE

FROM COMPANY;

SELECT \* FROM COMPANY\_VIEW;

DROP VIEW COMPANY\_VIEW;

ALTER VIEW foo RENAME TO bar;

**On our Tables**

Create view passenger\_view as select \* from passenger



Create view passenger\_going\_to as select \* from passenger where to\_place = ‘delhi’

