Database Design and Development

DATABASE DEVELOPMENT – VIEWS & STORED PROCEDURES

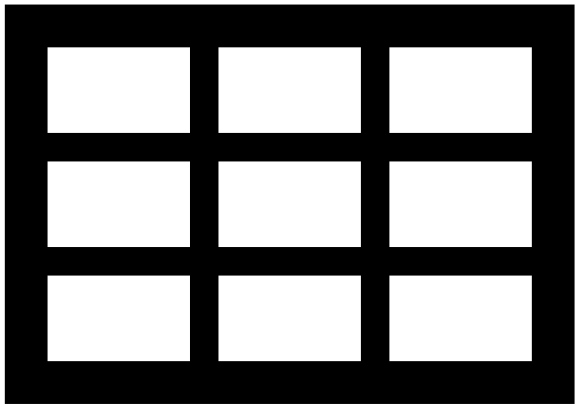
Views (Virtual Tables)

Concept of a view in SQL

- Single table derived from other tables called the **defining tables**
- Considered to be a virtual table that is not necessarily populated

Why use Views?

- Security - Views can be made accessible to users while the underlying tables are not directly accessible. This allows the DBA to give users only the data they need, while protecting other data in the same table.
- Simplicity - Views can be used to hide and reuse complex queries.
- Column Name Simplication or Clarification - Views can be used to provide aliases on column names to make them more memorable and/or meaningful.
- Stepping Stone - Views can provide a stepping stone in a "multi-level" query.



CREATE VIEW command

- Give table name, list of attribute names, and a query to specify the contents of the view
- In the first View, attributes retain the names from base tables. In the second View, attributes are assigned names

```
CREATE VIEW     emp_proj_view
AS    SELECT    e.SSN, e.fname, e.lname, p.name, ep.hours
      FROM      employee e JOIN employee_project ep ON e.SSN = ep.employee_SSN
                JOIN project p ON p.number = ep.project_number;

CREATE VIEW     dept_info (Dept_name, No_of_emps, Total_sal)
AS    SELECT    d.name, COUNT(*), SUM(e.salary)
      FROM      department d JOIN employee e ON d.number = e.dept_number
      GROUP BY  d.name;
```

Specification of Views in SQL

Specification of Views in SQL

View is always up-to-date

- Responsibility of the DBMS and not the user

DROP VIEW command

- **DROP VIEW** dept_info;

Retrieve data from VIEW

- **SELECT** name, hours **FROM** emp_proj_view **WHERE** fname = 'John';

Tables Vs. Views

A table consists of rows and columns to store and organized data in a structured format, while the view is a result set of SQL statements.

The table is an independent data object while views are usually depending on the table.

The table is an actual or real table that exists in physical locations whereas views are the virtual or logical table that does not exist in any physical location.

Views created from one or more than one table by joins, with selected columns. We can combine columns/rows from multiple table or another view.

Views reduces the effort for writing queries to access specific columns every time Instead of hitting the complex query to database every time, we can use view

Views can be used as security mechanisms by letting users access data through the view, without granting the users permissions to directly access the underlying base tables of the view. Views are created to hide some columns from the user for security reasons, and to hide information exist in the column.

It acts as abstract layer to downstream systems, so any change in schema is not exposed and hence the downstream systems doesn't get affected.

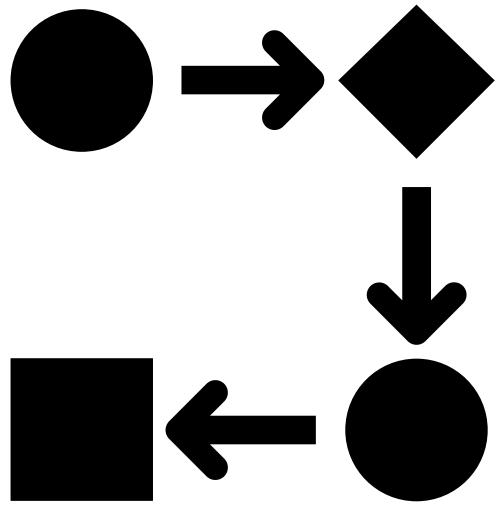
Stored Procedures

Concept of a Stored procedures in SQL

- Program modules stored by the DBMS at the database server.
- Stored persistently by the DBMS.

Why use stores procedures?

- Security
  - You can grant/revoke permissions on an individual stored proc.
  - Protect you from SQL injection attacks.
- Code reuse and encapsulation.
- When database program is needed by several applications.
- Reduce data transfer and communication cost between client and server in certain situations.



Declaring stored procedures:

```
CREATE PROCEDURE <procedure name> (<parameters>)
BEGIN
        <procedure body>
END
```

```
DELIMITER //
CREATE PROCEDURE selectAllEmployees()
BEGIN
        SELECT SSN, birthdate, gender, salary, address FROM employee;
END //
DELIMITER ;
```

Calling a stored procedure:

```
CALL <procedure or function name> (<argument list>);
```

```
CALL selectAllEmployees();
```

Database Stored Procedures

Database Stored Procedures

Each parameter has the following:

- **Parameter type**: one of the SQL data types
- **Parameter mode**: IN, OUT, or INOUT

```
DELIMITER //
CREATE PROCEDURE getEmployeeByName(IN first_name VARCHAR(15), IN last_name
VARCHAR(15))
BEGIN
	SELECT	SSN, birthdate, gender, salary, address
	FROM	employee
	WHERE	fname = first_name
			AND lname = last_name;
END //
DELIMITER ;

CALL getEmployeeByName('Franklin','Wong');
```

Views Vs. Stored Procedures

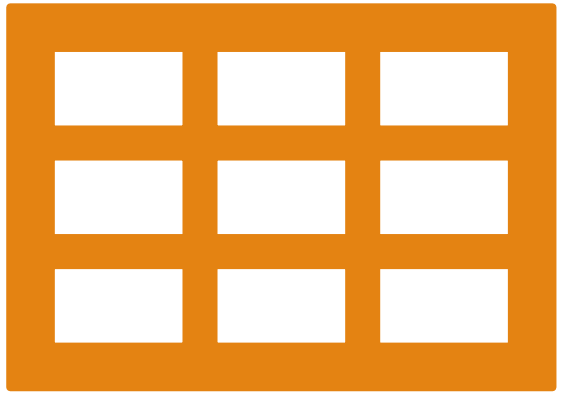
A SQL View is a virtual table, which is based on SQL SELECT query. A view references one or more existing database tables or other views. It is the snapshot of the database whereas a stored procedure is a group of Transact-SQL statements compiled into a single execution plan.

View is simple showcasing data stored in the database tables whereas a stored procedure is a group of statements that can be executed which contain variables, loops and calls to other stored procedures.

A view is faster as it displays data from the tables referenced whereas a store procedure executes sql statements.

Summary of SQL Syntax

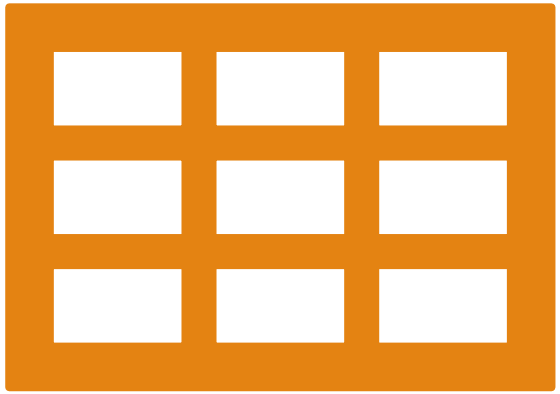
- CREATE TABLE <table name> (<column name> <column type> [<attribute constraint>]
  { , <column name> <column type> [<attribute constraint>] }
  [<table constraint> { , <table constraint> }])
- DROP TABLE <table name>
- ALTER TABLE <table name> ADD <column name> <column type>
- SELECT [DISTINCT] <attribute list>
  FROM (<table name> { <alias> } | <joined table>) { , (<table name> { <alias> } | <joined table>) }
  [WHERE <condition>]
  [GROUP BY <grouping attributes> [HAVING <group selection condition>]]
  [ORDER BY <column name> [<order>] { , <column name> [<order>] }]
- <attribute list> ::= (* | (<column name> | <function> (([DISTINCT] <column name> | *)))
  { , (<column name> | <function> (([DISTINCT] <column name> | *)) }))
- <grouping attributes> ::= <column name> { , <column name> }
- <order> ::= (ASC | DESC)



Summary of SQL Syntax

- INSERT INTO <table name> [(<column name> { , <column name> })]
  (VALUES (<constant value> , { <constant value> }) { , (<constant value> { , <constant value> }) }
  | <select statement>)
- DELETE FROM <table name>
  [WHERE <selection condition>]
- UPDATE <table name>
  SET <column name> = <value expression> { , <column name> = <value expression> }
  [WHERE <selection condition>]
- CREATE [UNIQUE] INDEX <index name>
  ON <table name> (<column name> [<order>] { , <column name> [<order>] })
  [CLUSTER]
- DROP INDEX <index name>
- CREATE VIEW <view name> [(<column name> { , <column name> })]
  AS <select statement>
- DROP VIEW <view name>

NOTE: The commands for creating and dropping indexes are not part of standard SQL.



References

Elmasri, R., & Navathe, S. (2017). Fundamentals of database systems (Vol. 7). Pearson

Nugent, D. (2017). Higher Nationals in Computing Core Textbook. Pearson Education Custom Content.