Indexes

Two Types:

1. Duplicate Index- Indexes that allows duplicate values for the indexed columns

2. Unique Index- Indexes that deny duplicate values for the indexed columns

An index can be simple or composite

Creating index

1.Simple :

Create index <index name> ON <table name> (<column name>);

2.Composite :

Create index <index name> ON <table name> (<column name1>, <column name2>);

Creating unique index

Simple :

Create unique index <index name> ON <table name> (<column name>);

Composite:

Create unique index <index name> ON <table name> (<column name1>, <column name2>);

Views

* A view is that it is stored only as a definition in system catalog.
* When a reference is made to a view, its definition is scanned, the base table is opened and the view created on the top of the base table.
*A view holds no data at all , until a specific call to view is made

Why views are created:
1. When data security is required
2. When data redundancy is to be kept to the minimum possible

Types of views

1. Read Only
2. Updatable

1. Read Only – used only to lookup data and nothing else.

2. Upadatable-For a view to be upadatable, it should meet the criteria

a. Views defined from single table

b. If the user want to insert record with help of view, then primary key column and all NOT NULL columns must be included in the view

c. The user can update,Delete records with help of a view even if a primary key column and not null are excluded from the view.

Syntax

Create View < view name> As
Select <column1> , <column2> from <table name>
Where <column name> = <Expression List>
Group By <Grouping Criteria> Having <predicate>;

Example:
Create view EMP_View AS select Fname, Mname, lname, dept from emp_mstr;

JOINS

- **INNER JOIN (simple join)**

Chances are, you've already written a statement that uses a MySQL INNER JOIN. It is the most common type of join. MySQL INNER JOINS return all rows from multiple tables where the join condition is met.

- **Syntax**
- SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;

JOINS

- **LEFT OUTER JOIN**

Another type of join is called a MySQL LEFT OUTER JOIN. This type of join returns all rows from the LEFT-hand table specified in the ON condition and **only** those rows from the other table where the joined fields are equal (join condition is met).

- **Syntax**
- SELECT columns FROM table1 LEFT [OUTER] JOIN table2 ON table1.column = table2.column;
- In some databases, the LEFT OUTER JOIN keywords are replaced with LEFT JOIN.

JOINS

- **RIGHT OUTER JOIN**

Another type of join is called a MySQL RIGHT OUTER JOIN. This type of join returns all rows from the RIGHT-hand table specified in the ON condition and **only** those rows from the other table where the joined fields are equal (join condition is met).

- **Syntax**
- SELECT columns FROM table1 RIGHT [OUTER] JOIN table2 ON table1.column = table2.column;
- In some databases, the RIGHT OUTER JOIN keywords are replaced with RIGHT JOIN.