## SQL

1. **What are RDBMS v/s DBMS…?**

**DBMS (Data base management system)**

* **Application store a data in file format**
* **There is no security (constraint) for data**
* **There is no relation between tables**
* **Ex:= DBMS file are XML**

**RDBMS (Relational data base management system)**

* **Application store a data in Table format**
* **There is security (constraint) for data**
* **There is relation between tables**
* **Ex:= RDBMS file are My sql, orcal**

1. **What are the data base languages..?**

* **Data definition language**
* **Data control language**
* **Data manipulation language(Tester will prefer)…insert , select, update, delete**
* **Transaction control language**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PRIMERY KEY AND UNIQUE KEYS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

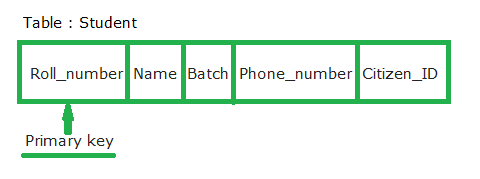
**PRIMERY KEY :-**

* **A primary key is a column of table which uniquely identifies each tuple (row) in that table.**
* **Primary key enforces integrity constraints to the table.**
* **Only one primary key is allowed to use in a table.**
* **The primary key does not accept the any duplicate and NULL values.**
* **The primary key value in a table changes very rarely so it is chosen with care where the changes can occur in a seldom manner.**
* **A primary key of one table can be referenced by foreign key of another table.**

**Example:-**

**For better understanding of primary key we take table named as Student table, having attributes such as**

**Roll number, Name, Batch, Phone number, Citizen\_ID.**

****

**Primary key values can never be reused. If a row is deleted from the table, its primary key may not be assigned to any new rows in the future**

**The roll number attribute can never have identical and NULL value, because every student enrolled in a university can have unique Roll\_number, therefore two students cannot have same Roll\_number and each row in a table is uniquely identified with student’s roll number. So, we can make Roll\_number attribute as a primary key in this case**

**UNIQUE KEYS:-**

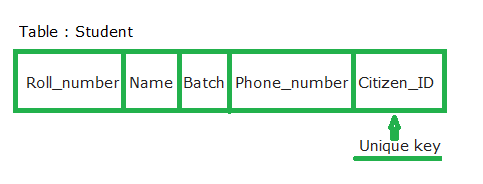
* **Unique key constraints also identify an individual row uniquely in a relation or table. A table can have more than one unique key unlike primary key.**
* **Unique key constraints can accept only one NULL value for column.**
* **Unique constraints are also referenced by the foreign key of another table.**
* **It can be used when someone wants to enforce unique constraints on a column and a group of columns which is not a primary key.**

**Example:-**

**Roll number attribute is already assigned with the primary key and Citizen\_ID can have unique constraints where each entry in a Citizen\_ID column should be unique because each citizen of a country must have his or her Unique identification number like Aadhaar Number.**

**But if student is migrated to another country in that case, he or she would not have any Citizen\_ID and the entry could have a NULL value as only one NULL is allowed in the unique constraint.**

**Unique key is same as primary with difference being the existence of null. Unique key field allows one value as NULL value. It won’t allow duplicate entries.**

****

**Key Differences Between Primary key and unique key:**

1. **Primary key will not accept NULL values whereas Unique key can accept one NULL value.**
2. **A table can have only primary key whereas there can be multiple unique key on a table.**
3. **Whereas unique key generates the non-clustered index.**
4. **Values in primary key columns can never be modified or updated.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Primary and foreign Key\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**foreign Key:**

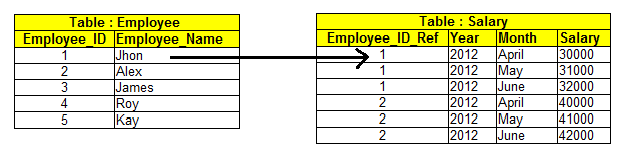
**When, "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table.**

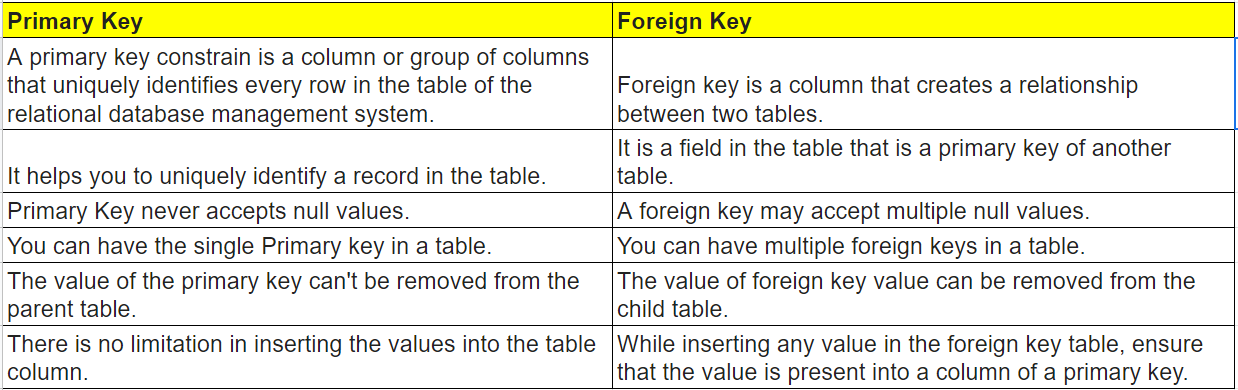
**In the example given below, salary of an employee is stored in salary table. Relation is established via foreign key column “Employee\_ID\_Ref” which refers “Employee\_ID” field in Employee table.**

**For example, salary of "Jhon" is stored in "Salary" table. But his employee info is stored in "Employee" table. To identify the salary of "Jhon", his "employee id" is stored with each salary record.**

**The advantage of using foreign key is that, the data is not getting duplicated. If foreign key concept was not there in RDBMS, entire info of an employee, such as First Name, Last Name, Id etc. had to be stored with each and every salary entry.**

**Another advantage of foreign key is that the editing master entry such as designation, address, etc. wont have any impact on the child table.**

****

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SQL COMMANDS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Where clause:-**

**Used for selecting the rows based on condition (Filtering the rows using where condition)**

**Use database name;**

**Choose database which is work on it**

Select \* from **table name;**

**Retrieve all the data from choose table**

Select \* from **table name** where **column name (salary)> digit value (30000);**

**table name + column name + conditional statement**

Select \* from **table name** where **column name (salary)<=digit value (30000);**

**table name + column name + conditional statement**

Select \* from **table name** where **column name(employ id) = digit(15);**

**table name + column name + conditional statement**

Select \* from **table name** where **column name is** null **;**

**Choose particular column which is value is** null **from table**

Select \* from **table name** where **column name (first name)= ‘string’;**

**Choose particular column which is value is string from table**

Select distinct **column name** from **table name;**

**Choose unique records from column**

Select distinct\*from **table name;**

**Choose unique records from table**

**With logical Operators (AND, OR, NOT)**

**1.**

Select \* from **table name** where **column name (salary)<=digit value (30000) And column name(employ id) = digit(15);**

**table name + column name + conditional statement 1 AND conditional statement 1**

**2.**

Select \* from **table name** where **column name (salary)<=digit value (30000)** OR **column name(employ id) = digit(15);**

**table name + column name + conditional statement 1** OR **conditional statement 1**

**3.**

Select \* from **table name** WHERE NOT **column name (first name)= ‘string’(‘david’);**

**Select all data from table which is not belong to devid**

**BETWEEN & IN Operators**

**Range:**

Select \* From **Tablename** Where **Columnname** Betweenint **value** And **intvalue;**

Select \* From **Tablename** Where **Columnname** Not Betweenint **value** And **intvalue;**

**Ex: Select \* From employee Where Salary Not Between 100000 And 300000;**

**Particular value point out**

Select \* From **Tablename** Where **Columnname** in **(intvalue, intvalue, intvalue, intvalue);**

Select \* From **Tablename** Where **Columnname** not in **(intvalue, intvalue, intvalue, intvalue);**

**Ex:- Select \* From employee Where Salary not in (10000,20000,30000);**

**Pattern Matching (Wild card characters)**

**Work ‘s with only “like” OPRATER**

**1. %.......> many characters(Percentile)**

**2. \_...........> Single Characters(Under Scroll)**

Select\*from **Employees** Where **First\_name** Like **’s%’…….First name column fetch a data which is start from S character**

Select\*from **Employees** Where **First\_name** Like **’%r’…….. First name column fetch a data which is Last character is r**

Select\*from **Employees** Where **First\_name** Like **’s%r’…… First name column fetch a data which is start from S & end with r character**

Select\*from **Employees** Where **First\_name** Like **’%s%’……. First name column fetch a data which is S character is somewhere in middle**

Select\*from **Employees** Where **First\_name** Not Like **’s%’…… First name column fetch a data which is Not start from S character**

Select\*from **Employees** Where **First\_name** Like **’%e\_’…… First name column fetch a data which is contains Before “e” Many Charaters but After “e” contain Only One character.**

Select\*from **Employees** Where **First\_name** Like **’\_\_\_’ …. First name column fetch a data which is contains only Three Charaters( Three Times Under Scroll)**

**Special One :-**

Select \* From **TableName** OrderBy **columname** DESC **;**

**EX:- Select \* From Table Alubims Order By AlbumId DESC;…….Arrange Album id column in descending order.**

**Select \* From Table Alubims Order By AlbumId ASC;…….. Arrange Album id column in Ascending order.**

**2.**

Update **Table Name** Set **Existing Column Name =’To be Replaceable column Name’** Where **Column Name = 348;**

**Ex:- Update Albums Set Title =’Test’ Where Album\_ID = 348;……. Update the column name + show particular column**

**3.**

Select \* From **Table Name** LIMIT **value;**

**Ex :-** Select \* From **Albums** LIMIT **10;……….Only fetch First Ten Value from table**

**4.**

**Scenario Where**

**We want to fetch data like Second highest Among the record We need to declear limit + 3-1+ record count(hw many ) for that we need Arrange A data in asecending oder And apply limit condition**

Select \* From **Table Name** OrderBy **column name** ASCLimit **n-1,1**

**Ex:- Select \* From Table Alubums Order By Album\_Id ASC Limit 2,1….result would be fetch Second Highest One Record Among the mention column**

**5.**

**Select MAX (column Name) From Table Name; …..**

Select **MAX (Album id)** From **Album……Fetch Maximum Value From Mentioned Column**

Select **MIN (Album id)** From **Album……Fetch Minimum Value From Mentioned Column**

Select **AVG (Album id)** From **Album……Fetch Average Value From Mentioned Column**

Select **SUM (Album id)** From **Album……Fetch Addition Of All Value’s From Mentioned Column**

Select **COUNT (Album id)** From **Album……Fetch Count Of Row’s From Mentioned Column**

Select **UPPER (Title)** From **Album……Convert & Fetch Upper Case Value From Mentioned Column**

Select **LOWER (Title)** From **Album……Convert & Fetch Lower Case Value From Mentioned Column**

**ODD AND EVEN DATA**

Select\*from **Table Name** Where **Column Name**

Select\*from **Employees** Where **Empoy\_id % 2 = 0;………..Fetch Even Data From mentioned Column**

Select\*from **Employees** Where **Empoy\_id % 2 = !0;…….. Fetch ODD Data From mentioned Column**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VARIABLE** | **EMP\_ID** | **NAME** | **MANGER\_ID** | **D.O.J** | **CITY** | **PROJECT** | **SALARY** |
| **0** | **121** | **John** | **321** | **1 Mar 16** | **Hyderabad** | **P1** | **20000** |
| **1000** | **321** | **David** | **986** | **1 Feb 17** | **Chennai** | **P2** | **35000** |
| **3000** | **421** | **Scott** | **876** | **1 Jan 16** | **Mumbai** | **P3** | **50000** |

**TABLE NAME = EMPLOYEE**

**Questions**

**1. fetch the data where salary of employee is more than 30000..?**

**SELECT \*FROM employee where salary>30000**

**2. choose a particular column which is value is null from table..?**

**SELECT \* FROM employee WHERE project is NULL**

**3. choose unique record from Column ..?**

**SELECT DISTINCT city FROM employee**

**4. fetch the data whose salary in between 20000 to 45000 range..?**

**5. fetch the data whose salary in between 20000 and 45000 range..?**

**6. fetch the data whose salary in between 20000 or 45000 range..?**

**7. fetch the data whose salary in between 20000 not 45000 range..?**

**8. fetch the data whose name not start with D..?**

**9. fetch the data whose name start with D..?**

**10. fetch the data whose name not end with D..?**

**11. fetch the data whose name has only three character ..?**

**12. fetch the data whose name start with three character and with d ..?**

**13. fetch the data whose name second character is d ..?**

**TABLE NAME = EMPLOYEE DATA**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EMP\_ID** | **FIRST NAME** | **Last name** | **Middle name** | **Job id** | **mangerID** | **DOJ** | **Salary** | **Commission** | **Dept ID** |
| 7369 | John | Smith | Q | 667 | 7902 | 17 dec 84 | 800 | Null | 20 |
| 7499 | Kelvin | Allen | F | 670 | 7698 | 20 feb 85 | 1600 | 300 | 30 |
| 7505 | Jean | Doyle | D | 671 | 7839 | 4 apr 85 | 2850 | Null | 30 |
| 7506 | Lynn | Dennis | V | 671 | 7839 | 10 jan 85 | 2750 | Null | 30 |
| 7507 | Leslie | Baker | B | 671 | 7839 | 22 feb 85 | 2200 | Null | 40 |
| 2521 | Cynthia | Wark | N | 670 | 7696 | 15may 86 | 1250 | Null | 40 |

**COMBINE TWO TABLES = EMPLOYEE DATA + DEPARTMENT WITH PRIMARY KEY (DEP ID)**

|  |  |  |
| --- | --- | --- |
| **DEPT\_ID** | **NAME** | **LOCATION\_ID** |
| 10 | Accounting | 122 |
| 20 | Research | 124 |
| 30 | Sales | 123 |
| 40 | Operations | 167 |