What is DataBases

Collection of information which is managed as per our requirement. To arrange data we use softwares these are called as DBMS(DataBase Management system)

SQL-Structured Query Language

If you want to arrange data in structured manner in tabular format

ORACLE----Licensed

SQL server ---Licensed

MySql----freeware

PostgreSQL--freeware

NOSQL--- unstructured format

JSON ---Javascript object notation, key-value,document

GraphDB----- graph format

linkedin, facebook

Memoery Databases:--- VoltDB,MemDB----- Backup server MYSQL

Research project ----size of data very small. faster retrieval is needed

To store the data to arrange the data we use data model

1. Hierarchical Model—(1968-1980) --- it stores the data in tree structure format
2. Networking Model --- in this data is stored in Network format
3. Relational Model ----- Data is stored in tabular format. All SQL databases uses Relational Data model to store data.

table, relation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a/c no | Name | Mobile | Email address | address | Balance | Type of account | Branch  code |
| 1 | Kishori | 1111 | Fddfgd | Aundh | 23456 | saving | 0001 |
| 2 | Kishori | 1111 | Fddfgd | Baner | 44444 | demat | 0001 |
| 3 | Kishori | 1111 | Fddfgd | Baner | 55555 | current | 0001 |
|  |  |  |  |  |  |  |  |

redundancy -----repetition of data

Normalization rules will help to design data model

Relational database --- Relation means table

account-customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a/c no | custid | Balance | Type of account | Branch  code |
| 1 | 10 | 23456 | saving | 0001 |
| 2 | 10 | 44444 | demat | 0001 |
| 3 | 10 | 55555 | current | 0001 |
| 4 | 9 | 444 | Saving | 001 |
| 5 |  | 5555 | saving |  |

Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| custid | Name | Mobile | Email address | address |
| 10 | Kishori | 1111 | Fddfgd | Aundh |
| 9 | Rajan | 222 | dsjfhjs | Baner |
|  |  |  |  |  |

---every row should be uniquely identified

---- sequence of rows and columns is not important.

Why to use database

------data is stored at centralized location, Hence sharing of data is possible

----- data is stored in structured format, relational format, retrieval data is very easy

-----because of relation, data can be stored in multiple small tables, so redundancy is reduced, so accuracy of data is more

-----Transaction controls----They support ACID property

What is Transaction

step by step procedure, in which either all steps should happen or none should happen

Transfer fund

1. src account check balance is sufficient
2. withdraw from source account
3. deposit the amount in destination account

Automicity – every transaction gets executed as a single unit

Consistency---- after every transaction data should be in correct state

Isolation----- any user when login should read the same data

Durability-----longer period of time consistency will be maintained by application

---------It is secure

SQL DataBase -------MySQL

It stores data in relational format, i.e in tabular format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a/c no | custid | Balance | Type of account | Branch  code |
| 1 | 10 | 23456 | saving | 0001 |
| 2 | 10 | 23456 | saving | 0001 |
| 3 | 10 | 23456 | saving | 0001 |
|  |  |  |  |  |

minimal subset of columns which identifies the row uniquely is called as primary key

in above table a/c number is primary key

Room booking table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Roomno | custid | Date of booking | amount | No of days |  |  |  |  |
| 1 | 100 | 1 Oct | 5000 | 3 |  |  |  |  |
| 1 | 100 | 1 Nov | 5000 | 3 |  |  |  |  |
| 2 | 100 | 1Nov | 5000 | 3 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Room

romno, room type,fscilites, rate

Types of keys in Databases

1. primary key --minimal subset of columns which identifies the row uniquely is called as primary key

---- every table should have only one primary key

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studid | Name | coursename | fees | Admission date | Adhar card number | Passport number | Mobile number |
|  |  |  |  |  | 12345 |  | 9822011111 |
|  |  |  |  |  | 12345 |  | 9822011111 |
|  |  |  |  |  |  |  |  |

1. composite key---if primary key is combination of more than one columns then it is called as composite key

(roomno+booking date)

1. candidate key---- columns which can probably become a primary key is candidate key

(passport num, adhar num, mobile num, stud id)

1. Alternate Key-🡪 all candidate keys which are not selected as primary key is called as alternate keys

(passport num, adhar num, mobile num)

1. foreign key--- a column which is dependent on another column for integrity(correctness) of data then it is called as foreign key

student-course

|  |  |  |
| --- | --- | --- |
| Stuid | cid | fees |
| 1 | C1 | 1111 |
| 2 | C2 | 2222 |
| 1 | C3 | 4444 |
| 3 | C2 | 3333 |

course details

|  |  |  |
| --- | --- | --- |
| Courseid | coursename | faculty |
| C1 | Java | Rohit |
| C2 | Spring | Sarika |
| C3 | .NET | Ashwini |

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Empno | Ename | Sal | deptno | projno |
| 111 | Shraddha | 12345 | 10 | 1 |
| 2222 | Prashant | 5678 | 20 | 1 |
| 333 | Shailesh | 3456 |  | 2 |

Dept

|  |  |  |
| --- | --- | --- |
| Deptno | Dname | Dloc |
| 10 | HR | Mumabi |
| 20 | Purchase | Pune |
| 30 | Accounts | Chennai |

Project

|  |  |  |
| --- | --- | --- |
| Proj no | Pname | duration |
| 1 | Inssurance | 3 |
| 2 | Loan | 4 |

student table

|  |  |  |
| --- | --- | --- |
| Sid | Sname | address |
| 1 | Rajesh | Aundh |
| 2 | Rekha | Baner |
| 3 | Ashu | Kothrud |

courseTable

|  |  |  |
| --- | --- | --- |
| Courseid | coursename | faculty |
| C1 | Java | Rohit |
| C2 | Spring | Sarika |
| C3 | .NET | Ashwini |

student-course

|  |  |  |
| --- | --- | --- |
| Stuid | cid | fees |
| 1 | C1 | 1111 |
| 2 | C2 | 2222 |
| 1 | C3 | 4444 |
| 3 | C2 | 3333 |

studid references sid of student table

cid references coursed of course table

1. super key--🡪 Any combination of columns which identifies the row uniquely is called as supper key

(studid,studid+name,studid+coursename, studid+coursename+name,………..)

1. unique key-----a column in which duplicates are not allowed. but it allows to store any number of null values.

(passport num, adhar num, mobile num)

SQL (Structured Query Language) MySQL

PLSQL(procedural language SQL)

if statement, loops, cursors

Type of SQL statements

|  |  |  |
| --- | --- | --- |
| Types of statement |  |  |
| DQL | Data Query Language—used to retrieve the data from table | Select |
| DDL | Data definition Language---it allows use to create or modify structure of table | Create, drop, truncate,alter |
| DML | Data manipulation Language----It allows us to manipulate data | Insert,update,delete |
| DCL | Data Control Language | Grant, revoke |
| TCL | Transaction Control Language | Commit, rollback, savepoint |

step 1 :---install mysql

<https://dev.mysql.com/downloads/installer/>

step 2: then use windows start button and select mysql command line

step 3: enter password

---for better organization of table create separate database

mysql>create database iacsd2111;

-----switch to iacsd2111

mysql>use iacsd2111;

--------download demobldmysql.sql and file store in folder d:\mysql

------to create emp, dept and salgrade table

mysql>source d:\mysql\demobldmysql.sql

----see the list of tables

mysql> show tables;

------see the list of all record in table

mysql>select \* from emp;

mysql>select \* from dept;

mysql>select \* from salgrade

-------see the columns in table

mysql> desc emp

mysql>desc dept

mysql>desc salgrade

------to clear the screen

ctrl+l

or

\system cls

--------to display only empno and ename column

select empno,ename

from emp;

--------to select only rows with empno=7369

select empno,ename

from emp

where empno=7369

------to select all employees with salary > 2000

select \* from emp

where sal>2000;

------in mysql default format for date : yyyy-mm-dd

select \* from emp

where hiredate>’1981-02-20’;

----to find all employees with deptno=10 and sal >2000

select \* from emp

where deptno=10 and sal>2000;

------to find all employees with sal >=1600 and <=2500

select \*

from emp

where sal >= 1600 and sal<=2500;

----to check the range use between … and

select \*

from emp

where sal between 1600 and 2500;

between and 1600 and 2500 values are inclusive

----------------to find all employees with sal not >=1600 and <=2500

select \*

from emp

where sal not between 1600 and 2500;

------ to find all employees with sal =1600 or sal =2500 or sal =3000

select \*

from emp

where sal=1600 or sal=2500 or sal=3000

-----use in operator

select \*

from emp

where sal in (1600,2500,3000)

------ to find all employees with sal is not =1600 nor sal =2500 nor sal =3000

select \*

from emp

where sal not in (1600,2500) or sal=3000;

-----to find all employees with commission null

select \*

from emp

where comm is null

in, between ..and, not in, not between, is null, is not null

--------to find all employees with name=’SMITH’

select \*

from emp

where ename=’SMITH’;

-------to find all employees with name starts with S

like operator is used for pattern matching

% ------ matches with 0 or more characters

\_ ----🡪 it matches with one character

select \*

from emp

where ename like ‘S%’

------to find all employees with name ends with N

select \*

from emp

where ename like ‘%N’

------to find all names which has A at second position

select \*

from emp

where ename like ‘\_A%’

---- find all employees which has R at 3rd position or starts with A

select \*

from emp

where ename like ‘\_\_R%’ or ename like ‘A%’

-----to find all employees with name starts with S and ends with H

‘S%H’

select \*

from emp

where ename like ‘s%h’

select \*

from emp

where ename like ‘s%’ and ename like ‘%h’

----all employees with name starts with s and t at 2nd last position and h is not at the last position

select \*

from emp

where ename like ‘s%t\_’ and ename not like ‘%h’

-----to find All employees whose name does not starts with A

select \*

from emp

where ename not like ‘A%’

----to find all emplooyes with name starts with either A or M or S

select \*

from emp

where ename like ‘A%’ or ename like ‘M%’ or ename like ‘s%’;

------to find all employees with name starts with any character within range s-z

This query is difficult to write using like operator hence better to use REGEXP

|  |  |
| --- | --- |
| **Pattern** | **What the Pattern matches** |
| \* | Zero or more instances of string preceding it |
| + | One or more instances of strings preceding it |
| . | Any single character |
| ? | Match zero or one instances of the strings preceding it. |
| ^ | caret(^) matches Beginning of string |
| $ | End of string |
| [abc] | Any character listed between the square brackets |
| [^abc] | Any character not listed between the square brackets |
| [A-Z] | match any upper case letter. |
| [a-z] | match any lower case letter |
| [0-9] | match any digit from 0 through to 9. |
| [[:<:]] | matches the beginning of words. |
| [[:>:]] | matches the end of words. |
| [:class:] | matches a character class i.e. [:alpha:] to match letters, [:space:] to match white space, [:punct:] is match punctuations and [:upper:] for upper class letters. |
| p1|p2|p3 | Alternation; matches any of the patterns p1, p2, or p3 |
| {n} | n instances of preceding element |
| {m,n} | m through n instances of preceding element |

----to find all employees with name starts with any character in range s to z

select \*

from emp

where ename REGEXP ‘ ^[S-Z]’

-----to find all employees with name starts with A and ends with N

‘A%N

% ----🡪 .\* in REGEXP

\_ ----🡪 . in REGEXP

select \*

from emp

where ename REGEXP ‘^A.\*N$’

or

select \*

from emp

where ename like ‘A%N’

‘^A.\*N$’

ALLEN --- this will match

ALLENWAR –this will not match

‘^A.\*N’

ALLEN --- this will match

ALLENWAR –this will match