DBMS

Lets upgrade

MySql.

Day 1

application

Front end

html, CSS GUI/UI

Interactions

↓

done for creating a webpage. Here we write code in a language.

back end

API

↳ coding language means collection of functions.

Database

↓

where all the data is stored and managed.

↓

there are many softwares available for creating and storing datas.

* The front end language are so simple so they cannot interact with the database directly. So another one layer is present.

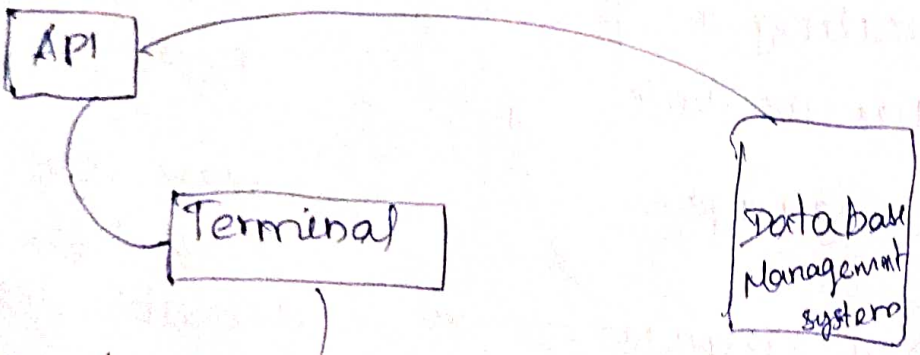
* UI will request to API to send the data from the database. So API fetches it and send them to the front end webpage where things are displayed.

* Free - Open source database system - My SQL.

* For a single application there can be multiple databases.

process of database:
1) Server is created
2) Databases are created.
3) Datas are installed in the datab

are done using writing commands in the terminal
↳ in which language the commands are written
SQL - Structured Query Language.



API - backend of real - applications.

eg: Mysql workbench

* Comment : -- in SQL.

* Datas in the database are stored using tables

MySQL - Database system software which gives us a server and inside which we connect to it & create databases, & commands which is then

connectus to Applications.   Day 2

* desc - describe (x) -> describe a table.

↳ command used to show the table and the details in them.

show table:

↳ used to show the table.

↳ show all the tables from the MY SQL

show the details of the particular table.

Syntax: desc tablename;

* table = entity
* Columns = attributes
* rows = data / tuples ↳ collection of data or single record.

---

* View the data present in the table we use

select * from products:

↳ is a clause is used with Queries to enhance the power of the query.

from clause is used in Oracle compulsory but in MY SQL it works without that also.

Inserting a data into the table

Syntax:

Insert into tablename values (the datas needed (' '), ' ' , ' ' ← to inserted as per oder);

(enter the strings in this (' ') format)

* Extra garbage data means Extra memory space then Extra cost for the Extra space of data.

↳ to make a table without these are known as Constraints which are rules that apply on columns → this makes what kind of data is allowed and what kind of data is not allowed.

* Constraints are 6 available in SQL.

1) not null
2) unique.
3) Default
4) check
5) primary key
6) foreign key.

these are included in the queries which makes the limit from entering less data

drop - deops the entire table and the datas in them completely (permanent) → and the database

delete - deletes the particular record in the table or whole table but not the structure.

Truncate - delete the record and the table in the database ~~temporarily~~ without deleting table structure.

↳ that is headings of the column remain but the datas inside that will be deleted.

1) not null - the data inside the column cannot be null.

2) primary key - always unique and it will never allow null. (same name cannot be repeated twice).

↳ mostly ids will be null.

3) unique - the name or data will be unique two rows cannot have same data. → can be null.

→ is a combo of unique & not null.

4) check - used to check a data.

eg: check (age > 12);

5) Default - if any data is not entered then the default value will be entered.

eg: default "1";

* auto_increment keyword - is used for automatically increasing the id number with regular order.

↳ so that we no need to enter the column names data.

Eg: 1 // forcefully (10) → 5 lost → so next box (11) //

if we forcefully entered data for that row then the particular number is lost and for the next row the particular numbers next no comes.

* But when the inserted data is wrong the auto increment no is also lost then the next query the autoincrement tries to skip that particular no and work with next no.

---

* Create table :

create table tablename (colname datatype, col ....);

eg: create table users (name varchar[20], age int);

* Display all datas :

Select * from tablename;

eg: Select * from users;

* Insert datas into table :

insert into tablename values (' ', ...); ← String means

eg: insert into users values ('Ranjani', 20);

Day-3

* Select only particular columns in the table

Select col1name, col2name from tablename;

eg: Select name, age from users;

* Using as clause

Select col1 name as ____ , col2 name as ____

from tablename;

eg: Select name as firstname, age as entry from users;

*row filteration - clause are used.

Eg: where, Having

Select [*/columns] from tablename where condition;

eg: select * from products where quantity >= 2;

all relational operators works.

* update a record:

update tablename set colname = new value where condition;

more than one condition can be in pattern: → using conditional operators and, or

*Always where condition works with more than one condition using and, or.

eg: Select * from users where age >= 20 and age <= 40;

Select * from tablename wherecondition:

(or)

Select * from tablename where colname between condition1 and condition 2;

eg:

Select * from users where age between 20 and 40;

*in, not in, like

Select * from tablename where colname in ['', '', '',

*Same for not in → the condition that have alone other than that all will be executed.

*like :

pattern matching is done u
← like op

Select * from tablename where colname like condition;

eg: Select * from users where name like 'A(1.)';

↓

after that anything fine

*Sorting or order by :

select * from tablename order by columnname;

eg:

select * from ~~tabl~~users order by age desc;

↳ descending

asec;

↳ ascending.

*Format of a sql select query

Select [*/columnname] from tablename (1) where condition (1 or more) (2) group by (3) having (4) order by (5) limit

*limit clause

Select * from tablename limit condition;

eg: select * from users limit 3;

↳ first 3 rows are displayed.

Select * from users limit (3), 5;

↳ The 3rd record is skipped and then first 5 records are displayed

as one skipped the 6th record

*delete the record:

delete from tablename where condition;

eg: delete from user where id = 11;

* remove all the data from the table.

delete from tablename;

*delete all the records but the auto increment is present in the table only and the table is not deleted. and the table is not resetted.

* Truncate - the records are deleted and the table is resetted, and the table will be present. (all the records of the table is deleted.)

* drop - drops the whole database and the table

Day-4

* foreign key - is the columnname which is from another table used to connect two keys and link the datas of the table

* This foreign key should be the primary key in one table which is used to connect both the tables where the primary key treated as foreign key in the another table.

* One foreign key from one table is taken to the major table.

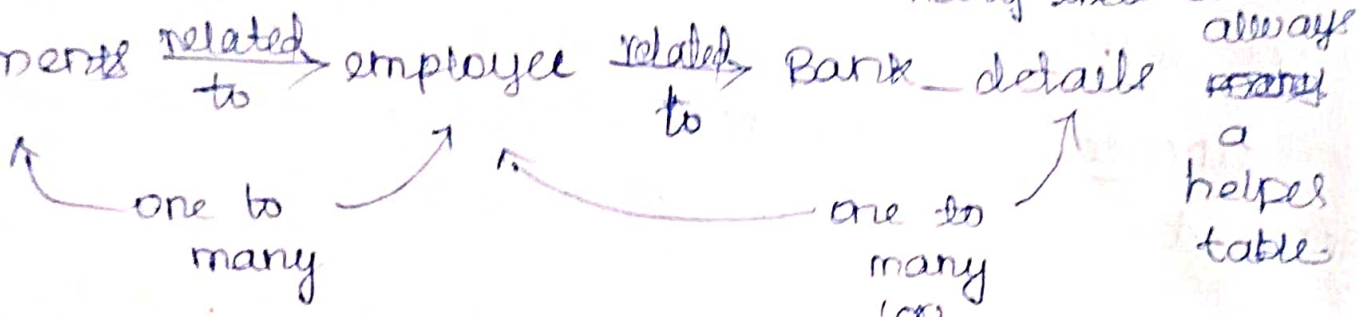
* But in one table there can be multiple foreign keys.

* In two tables who is giving the reference are known as parent that is who gives the foreign key is parent and who uses that foreign key are child.

Types:

1) One to One 2) One to many 3) many to many
(or)
in many to many there are always a helper table.

Relationships:



Day-5

* Alter command is used to add a column in the table.

Alter table tablename add colname datatype;

* Aggregate functions:

Min max count sum Avg

↳ Select MIN (column_name) from table_name where condition.

Select MAX(column_name) from table_name

where condition;

count → count the rows.

select count(*) from tablename; → count all the rows in the particular table. (in this null values are also counted)

select count (column_name) from tablename where condition;

Sum:

↳ total sum

Select SUM (colname) from tablename;

Avg:

↳ average value:

Select AVG (colname) from tablename;

SQL Aliases:

↳ used to give a table or column in a table a temporary name.

Select colname As newname from table;

Groupby:

Select colname from table name where condition group by colname;

* having clause is used because where keyword cannot be used with aggregate fns:

Select colname from tablename where/having condition;

* Joins: Whenever we want to fetch data from two different tables in a single result set you need to use join.

* Union: Combining two different queries.

→ types: A types: 1) inner join 2) outter join
3) right join 4) left join
5) cross join. 6) equi join

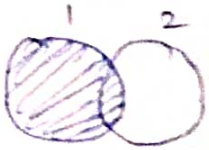
* Joins are known as to fetch data from multiple tables based on a relation.

Syntax:

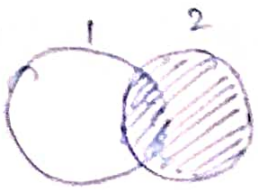
Select * from table1 (or) left table (which join mention that) inner join table2 (or) right table on relationship.

Inner join:

Left join:



Right join:



Full join: → cross join

→ multiplication of both the tables

Self join:

↳ a table is joined with itself.

equi-join: ↳ like inner join.