⇒ Database :- A database is an organized collection of data.

⇒ Database Archit:

a) Two-tier → user & application ↙ network to database system
b) Three-tier → user & application to application server & database system

⇒ Different types of keys:-

a) Super key → A superkey is a combination of columns that uniquly identifier any row within a RDBMS

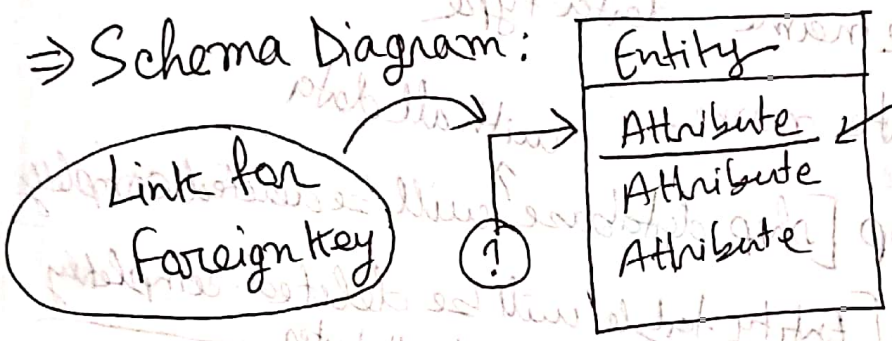
b) Candidate key → Super key is a candidate key if it is minimal.
→ all minimum posible key of superkey to indentifey any row.

c) Primary key → One of the candidate keys is selected as the primary key.
* can't be null value
* can't repeated same value

d) Foreign key → Value in one relation must appear in another.

⇒ Referencing relation → যে Refer করে [2nd table]
⇒ Referenced relation → যাকে Refer করা হয় [1st & main table]

⇒ Schema Diagram:



DDL ← Data Definition Language

⇒ Create table → Create table Entity(
Attribute name data type,
Attribute name data type);

⇒ Set Primary key → Primary key (Attribute name)

⇒ Set Foreign key →
Constraint fk_abc foreign key (Attribute) references
Constraint → key word
fk_abc → different for different f.k
foreign key → key word
(Attribute) → Referencing table A
references → key word
on update cascade on delete cascade

Entity (Attribute)
Referenced → Table name
Attribute → Attribute name

~ Any changes in primary key value it will ~~effect~~ be changed in foreign key value.

⇒ Alter :-

Alter table Entity

for add ← Add Attribute name · data type

for delete ← Drop Attribute name

→ Drop database shop [shop database will be deleted completly with all data]

→ Drop table Entity [Entity table will be deleted completly]

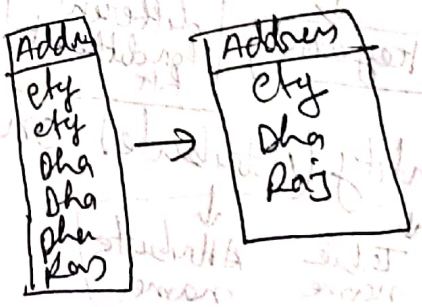
DML ← Data Manipulation Language (just values not attributes)

⇒ Insert → Insert into Entity values (—, —, —); (values & use '' for string.)

⇒ Select Query → Select A1, A2, A3, —, An
from Entity
where condition ← Ex: where id = 1300 [use '' for string]

→ Distinct keyword [to show only different values form a column.]

Ex: Select Distinct Address
~~from~~ from customer



⇒ Multiple condition for select:

→ where address = 'Dha' or address = 'ctg'

→ where class = 7 & section = 'A'

⇒ Select all columns by a ~~sign~~ single statement :-

Select * ← key word

⇒ Select + calculation:

Select P_id, (quantity * selling price + vat),

~~Rename~~

⇒ Rename operation:-

(quantity * selling price + vat) as total_price

← keyword.

⇒ update:

update Entity
Set Attribute name = value
where (condition);

⇒ Delete:

Delete from Entity
where (condition);

⇒ String operation:-

Operator ⟶ like

⇒ special characters:

% ⟶ Matches any substring

_ ⟶ Matches by single character

⇒ Pattern Matching example:

'Mr%' ⟶ Matching any string starting or beginning with "Mr"

'%abc%' ⟶ Matching any string containing "abc" as the substring

'___' ⟶ Matches any string of exactly three character.

'___%' ⟶ Matches any string of at least three characters.

string >= 3

⇒ Ordering the display of row / Tuples:

→ operator ⟶ order by

→ assending ⟶ asc.

→ dessending ⟶ desc.

~~select attribute~~
~~from Entity~~
~~where attribute like~~
~~'condition';~~
~~order by~~

select attribute
from Entity
order by attribute asc

⇒ Between operator:

~~Select product id from~~
Select * Entity

⇒ Between operator: [Must use AND operation]

Select Attributes
from Entity
where Attribute between value condition and value condition

or

where condition and condition

⇒ Join Query:

Select Entity.Attribute
from those Entities
where condition (with Entity.Attribute)

⇒ Aggregate functions:

→ use in select query

- avg (Attribute)
- max (Attribute)
- min (Attribute)
- sum (Attribute)
- count (Attribute) ← null কেও count করে। all null then return 0.

Ex:
select count (order-id)
from customer-order
where orderdate = '2010-10-01'

⇒ Group by:

রেকর্ড বিভিন্ন group by হয়। (same)

select Attribute, count (Attribute)
from Entity
group by Attribute ← রেকর্ড বিভিন্ন group by হয়

→ Group by ব্যবহারের ক্ষেত্রে condition দিতে "having" ব্যবহার করতে হয়।

Ex: having count (Attribute) > 2.