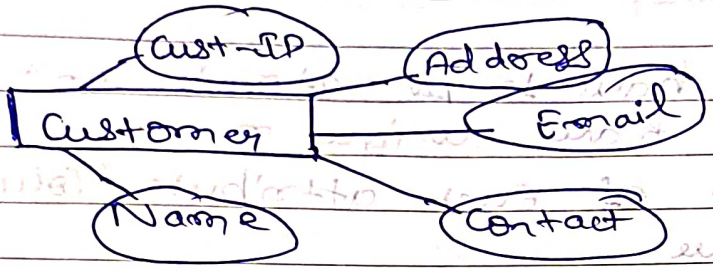
Relational Model



- We change the above ER-model into tabular form and attributes are changes into columns

Index	Cust-ID	Name	Address	Contact	Email
1	10	ABC	---	---	---
2	11	DEF	---	---	---

Customer

- Each row entry is called a typle.
- Degree of table → No. of attribute.

Cardinality → total no. of typle.

→ ER model → Relational model.
For each entity we made an table.

Q) Why we use relational model?
→ Firstly we made a ER model of the DB and then we made a Relational model to implement a DBMS Software that we called RDBMS.
E.g. ⇒ RDBMS = MYSQL, MS-Access, ORACLE

→ Important properties of a table in RM.

- ⊙ The values have to be atomic. Can't broken further.
- Each table have unique name.
- ⊙ The name of each attribute / Column must be unique.
- ⊙ Each tuple must be unique in a table.
- ⊙ The sequence of row and column has no significance.

→ Relational Model keys

- A set of attributes which can uniquely identify an each tuple.

① Super Key: (SK) Any combination of attributes present in a table which can uniquely identify each tuple.

- It can be NULL.

Eg. taking attribute of previous customer table.

[{Name, contact}, {Name, email},
{cust-ID, email}, {cust-ID}
{cust-ID, Name, contact, email}]

(2) Candidate Key (CK) :- A minimum subset of super keys, which can uniquely identify each tuple. It contains no redundant attribute. It shouldn't be NULL.

E.g. [{cust-ID, contact}, {cust-ID}, {cust-ID, Email}]

- Here we remove the "name" attribute as it doesn't uniquely identified and two customer can have same name. Although (name, email) may can uniquely identify customer.

(3) Primary Key (PK) :- A subset of CK, which has least no. of attributes.

E.g. {cust-ID}.

(4) Alternate Key (AK) :- In a subset of "CK" we choose only "cust-ID" as a primary key. All the other keys are AK.

(5) Foreign Key (FK) :-

- It shows relation between two tables.

E.g. -> We use two table to show the relationship. there "customer places order."

Customer :- (cust-ID, name, address, contact no.)
order :- (order-ID, timstamp, delivery date).

- Here we take primary of customer table and put it as a attribute in the order table. This attribute is called foreign key as we can find it which person has order.

- Here Customer table is "Parent table" and order table is "child table."

(6) Composite key :- Primary key formed using at least 2 attributes.

(7) Compound key :- Primary key formed using 2 foreign key.

(8) Surrogate key :-

- Synthetic Primary key, generated automatically by DB & usually an integer value.
- May be used as P.K.
- E.g. => Suppose we have two schools data & we are merging it. Both schools have different primary. So we use S.K.

School A

reg.no	name
101	ABC
102	DEF

School B

reg.no	name
B101	MNO
B102	PQR

→ Merged.

SK.	reg.no	name
1	101	ABC
2	102	DEF
3	B101	MNO
4	B102	PQR

→ Integrity Constraints

CRUD (Create, Read, Update, Delete) operations must be done with some integrity policy so that DB is always consistent.

① Domain Constraints

- Restricts the value in the attribute of relation.
  E.g. ⇒ We want to specify that the student DOB should > < 2002.

② Entity constraints

- Every relation should have PK., PK != NULL.

③ Referential Constraints

- Foreign key must have the matching Primary key for its each value in the parent table or it must be NULL.

④ Key constraints

① NOT NULL :- restrict the user from having NULL values.

② UNIQUE :- ensures that all values are different.

③ DEFAULT :- Set default value, if no value is entered then use the

④ CHECK :- It keeps check that integrity of data is maintained.

⑤ Primary key :- This attribute or set of att. uniquely identify each tuple

⑥ Foreign key :- Shows relation between two table.