DBMS - 01

Mantra
↳ Always try to go for dummy tables if stuck somewhere

Date: 25/02/2025

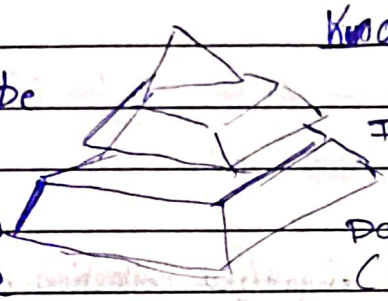
Introduction:

(1) "Data refers to raw, unprocessed set of facts and figures that we collect".

(2) Information → Information answers questions like who, when, where & so forth

Logically related structured data

(3) Knowledge: Information can be synthesized based on business choices to obtain actionable insights. This is called knowledge.



eg: Top performers (90 marks)

eg: A Grade

* Some features of the application in terms of data:

- Data **Integrity**: Data should be accurate, consistent and reliable.
  (Social media profile should contain a valid country name)
- Data **Retrieval**: Data should be available and usable.
  (Social media user should be able to access their data all the time)
- Data **Security**: Data should be protected from unauthorized access.
- Data **Independence**: A particular functionality should not depend on any specific application.
- **Concurrent Access**: Application should support multiple users and processes similtaneously without voilating data Integrity.

* **Flat Files :** Excel files, Word files, JSON, XML etc.

Traditionally data is stored in flat files

Date: / /

* Can be accessed using a programming language

Limitations :

Data Redundancy

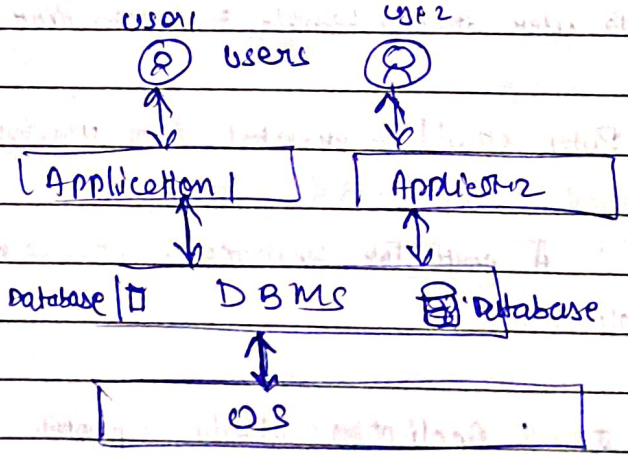
Dependency of program

Lost of data

Scalability challenges.

* **Database :** Structured and organized collection of logically related data and description of these data, designed to manage, retrieve and update information as per the needs.

* **DBMS :** Software Application for controlled access to the database.



Quiz1 - Intro ★★

ACIDS → Reliable, Consistent, Integrity

Date: / /

Quiz

★ which come first ??

★ Types of Database Systems:

Relational Model : stores data in Relations : i.e Tables (Attributes - values) (Property)

Employee (id, Fname, Salary, Bonus)

★ Attribute → Column name

★ Rows / Records / Tuples : → rows → data

★ Cardinality → No. of rows.

★ Degree → No. of columns.

★ NULL → Null does not have any Data Type

Represents a value of an Attribute that is absent/unknown ★

↓

"Absence of Data" (unknown)

★ domain

↓

"Set of all possible values that" an attribute can hold

Bonus can have values >=0 but

★ Pure mathematics can't be applied not like ∞ 😊

only largest possible value of that data type.

Example : Domain of Dept → { HCP, ETA, IVS }

only

RAJDHANI

Date: / /

* Quiz :

Q1. Relational Database is a collection of:
Tables/relations.

Q2. A relationship among a set of values in a table is represented by:
ROW
represents the collection of interrelated data.
(like the row 2 represents the whole related data of Employee 2 so that data is related to Employee 2)

Q3. The set of permitted values
↓ Domain

Q4. NULL (unknown value)

Q5. Cardinality – 4

Q6. Degree – 3

4) If customs can have multiple accounts and multiple shared.
Then

2	1173	Both repeating but combination is unique
2	1174	
3	1173	

So here {CustNo, AccountNo} both makes the composite P.K.

RAJDHANI

* Data Integrity and Constraints

↳ refers to accuracy, consistency, and reliability of data throughout its life cycle.

* Entity (Actually a table also ↳ one Tuple) represent that only.

Date: / /

* Can be ensured through CONSTRAINTS which are used to restrict data that can be entered. → Table on Next

How acc, consi, reliabi achieved through Entity Integrity.

[Imp] → { } written in curly braces

* Candidate Key: → identified during the design of Database only
2) based on [Business Logic] and needs.

1) "Minimal set of Attributes (columns) that can uniquely identify a Tuple (row) in a relation (table).

Examples:
- EmployeeNo — ✓
- AdharNo — X
- Name, DateOfBirth — ✓ (for small group)
- Salary X
- EmployeeNo, DateOfBirth → X because EmployeeNo itself is unique (see definition)

it can be a PRIMARY KEY

* Example 2: Accounts in Bank AccountDetails (AccNo, CustNo, name, Type)

1) If one customer can have only one account and account cannot be shared:
Both {AccountNo} and {CustomerNo} possible CKs

2) If a customer can have multiple accounts i.e.

1 → 1173
1 → 1125
1 → 1723

→ In this case {AccountNo} only is the possible CK.

3) If Cust can only have 1 acc. but can be shared. Repeating CustomerNo

AccNo	CustNo
1173	2
1173	3
1173	4

Here accountNo will repeat. ←

→ Here only {CustId} can be a possible CK.

RAJDHANI

* PRIMARY KEY (UNIQUE, NOT NULL)

Primary v/s Candidate Key

↓

"chosen specific Candidate key to uniquely identify a row/tuple.

Date: / /

↓

* Cannot have null
* should have small/short size

* Composite Primary Key.

↓

* Ensures Integrity by preventing duplication of data, and
* Reference for relationships with other table

PR. → {EmployeeNO}

* UNIQUE KEY : • can allow null values.
• Set of one or more columns whose values is required to be unique

* foreign key : → (can contain NULL values) (Next page more detail)

↓ UNIQUE KEY

* Primary key of another (Parent) used in this table

table for the reference.

(referencing) Child ← relationship — Referenced

* Double clarification : Both Candidate Key and Primary Key can be used for data Integrity by preventing Duplication of data.

1) 1 Table can have → only 1 P.K.
2) Candidate keys can be multiple

RAJDHANI

★ Qw3 – Data Integrity

Employee

Date: / /

EmpNo	Name	Aadhar	Salary	DOB
1	AKS	12345	8K	
2	BKS	34567	2K	
3	ABC	45678	20K	
4	XYZ	56781	20K	
5	Aditya	78190	35K	
6	Aniruddh	19235	35K	
7	Gaurav	23456	3K	
8	KEF	34901	3K	

tuples

unique

Candidate – Min set of columns → row

PK – UNIQUE NULL X → row

Consistent over a period of time

Size should be small

Candidate Key
{empNo}
{Aadhar}
{Name, DOB} {Name, DOB}

Primary key based on ★ Criteria

empNo attributes y Primary key

Mandatory
- Must uniquely identify a tuple.
- Null X

Desirable
- Should not change with time
- Should have short size. eg numeric.

All CK → PK ✓
All PK → CK

★ Composite Primary Key :

When two or more columns together identify the unique row then it is referred as Composite Primary Key.

If we have chosen (Name, DOB) as Primary Key in Employee table then it is Composite.

★ Candidate Keys me se kon-si PRIMARY KEY leni hai → Depend karta Apne pe karta.

* Quiz – Data Integrity

Date: / /

Employee

EmpNo	Name	Aadhar	Salary	DOB
1	AKS	12345	8K	—
2	BKS	34567	2K	—
3	ABC	45628	20K	
4	XYZ	56781	20K	—
5	Adithya	78190	35K	
6	Anirudh	19235	35K	
7	Gaurav	23456	3K	—
8	KEF	34901	3K	

Candidate - Min set of columns → row
PK - UNIQUE NULL X → row

Consistent over a period of time
Size should be small

Candidate Key
{EmpNo}
{Aadhar}
{Name, DOB} {Name, DOB}

Primary key based on → EmpNo attributes of Primary Key

* Criteria

Mandatory	Desirable
• Must uniquely identify a tuple.	• Should not change with time
• Null X	• Should have short size. eg numeric.

All CK → X PK
All PK → CK

* Composite Primary Key :

When two or more columns together identify the unique row then it is referred as Composite Primary Key.

If in Employee table
If we have chosen
(Name, DOB) as Primary Key
then it is Composite.

* Candidate Keys me se koi si
ek PRIMARY KEY
bani hai →
Depend karta
Apne pe karta.

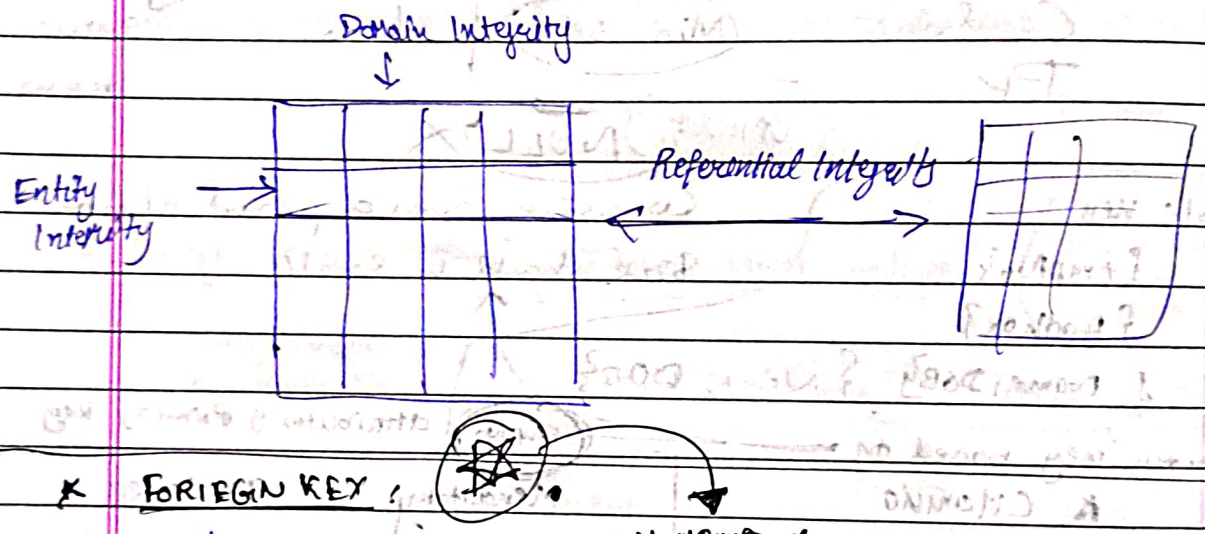
RAJPUTANI

Data Integrity and Constraints :

ensured through constraints.

Date: / /

Integrity Type	Definition	Enforced through
Entity Integrity	row/Tuple (must be unique in itself) (prevents duplication)	PRIMARY KEY
DOMAIN INTEGRITY	column/attributes	Data Types & CHECK CONSTRAINT
REFERENTIAL INTEGRITY	b/w relations (2 or more)	FOREIGN KEY



FOREIGN KEY :

↓

When one table uses the primary key of other table for reference.

↓

- That column in child table (Referencing table) is called its FOREIGN KEY.
- FOREIGN key in the child table can contain NULL values.

UNIQUE Key or

Trick. ★

★ Self-Referencing Foreign Key

A foreign key of a table can reference its own table's PRIMARY key

★ KEY CONFLICT Problem :

We can Rename the PRIMARY KEY / UNIQUE column being referenced in the Parent table without any problem.

RAJDHANI

Can use only those computers that are available in the computer table.

Uses A

Date: / /

Employee Table : Computer table

id	ename	dept	compId	← Relationship	compId	name	model
			1001		1001		
			NULL		1002		
			1002		1003		

child/Referencing Table — Parent/Master/Referenced Table.

- Can be NULL
- Can be duplicates.

* numerous rows in the child table can refer to the same row of the parent table, depending upon the type of the relationship.

* Quiz - Data Integrity

Q1. Relation b/w two tables → Foreign key.

Q2. only 1 P.K and Although not desired but Strings/char can be PKs.

Q3. custId

Ans. RechargeId (Dhyaan se padho → Niche vaale table Ka pucho)

Q4. (Operator, Plan) → Unique and Not Null

Q5. FK → 1

* Exercise - Data integrity and controlling.

Table 1 : Trainer (TrainerId, TName, EmailId, ContactNo)

CKs -> {TrainerId}, {emailId}, {contactNo}

Date: / /

PK -> {TrainerId}

FK -> { } -> No foreign key

here it is null

Table 2 : Member (MemberId, Name)

CKs : {MemberId}

PK : {MemberId}

FK : { } -> NO

Table 3 : WorkoutType (WorkoutTypeId, Level)

CKs : {workoutId}

PK : {workoutId}

FK : {

Table 4 : WorkoutSessions (TrainerId, WorkoutTypeId, MemberId, Tm

CKs : { TrainerId, workoutTypeId, MemberId }

PK : { same }

FK : {TrainerId}

{workoutTypeId}

{MemberId}

(Read the question carefully and get insights on what can be multiple keys there)

* Exercise 2: Keys (BookStore)

- Author(AuthorId, AuthorName)
  - CK = {AuthorId}
  - PK = {AuthorId}
  - FKs = { }

- Book (BookId, ISBN, Status)
  - CK : {BookId} -> multiple
  - PK : {BookId}
  - FK : { ISBN }

- BookDetails (ISBN, BookTitle, AuthorId)
  - CK -> {ISBN}, ~~{BookTitle}~~
  - PK -> {ISBN}, ~~BookTitle~~
  - FK = { AuthorId }

-> Ek particular Book type ki baat ho rahi.

ye ek unique hai key nahi

same

key nahi ho sakti because there can be multiple books with same title

* 1 Employee can have only 1 manager

* Relationship hota hai b/w entitie → us Relationship ka Cardinality hota hai

Date: / /

Junction Table — When 2 or more tables are connected with the help of the 3rd table

structure

V. V. Imp

* **Relationships** : associations of one-Entity with another entity through a **key**

to link related data for efficient querrying, and data integrity.

Each relationship has a **name**,

ex: A compatter is allocated to an employee.

* Mantra

Yaha hamesha 2-way relationship hoga.

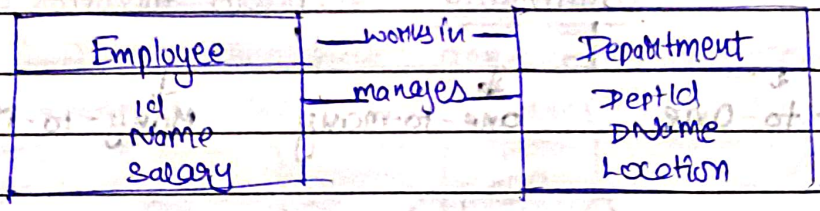
* there can be more than 1 relationships b/w entities.

Imp: Next part will be complex

e.g.

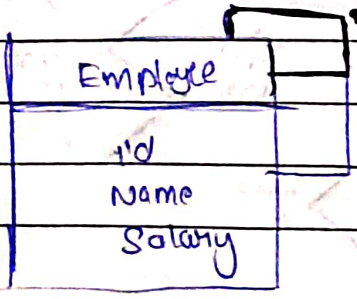
An employee works in a department,

An Employee also manages the department



* A relationship can also exist between instances of the same entity

↳ ex: An employee reports to a manager



Computer

Always work on this

Date: / /

* Cardinality of Relationships:

Cardinality of relationship is the number of related records or entities that can be associated with each other record or entity in the other table.

* Employee and Computer:

1) How many computers can be allocated to an Employee?
2) Can computers be shared among employees.
3) Can employees exist without being allocated a computer.

If 0 or 1 computer can be allocated 0 or 1 employee then the cardinality of relationship b/w these two entities is

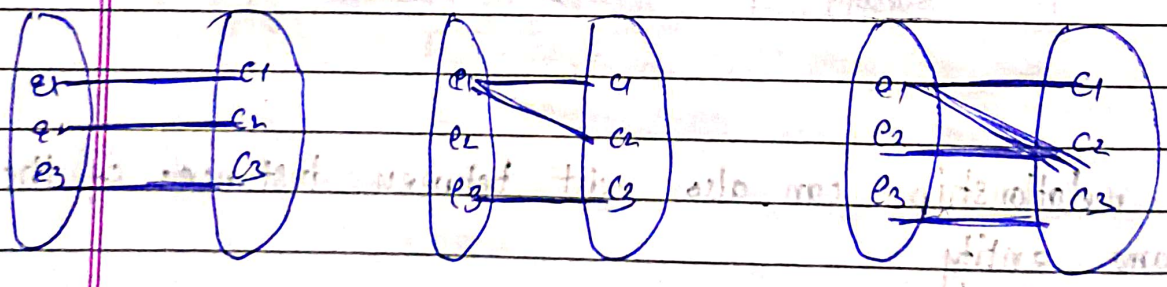
1:1

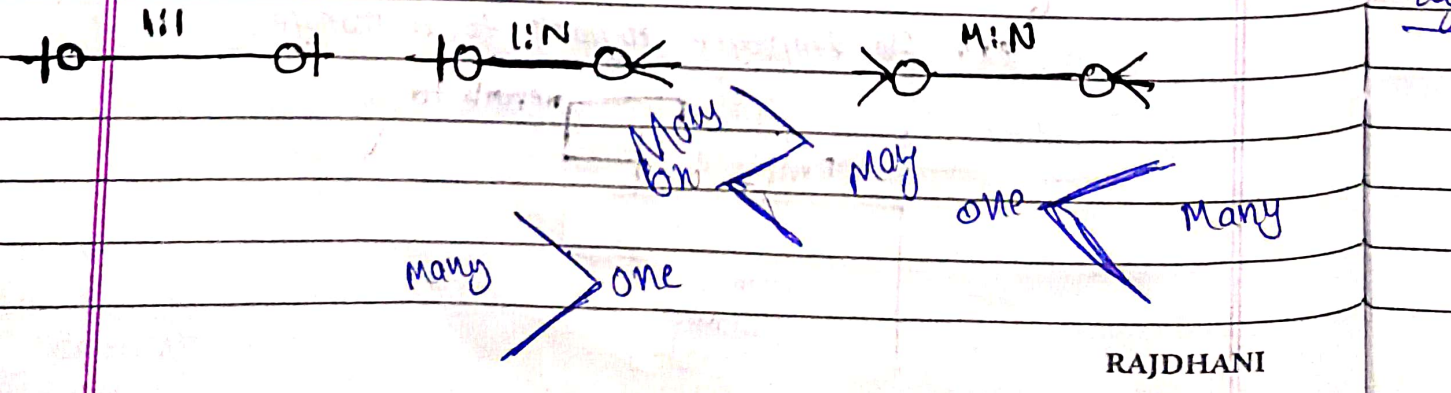
Mantra

Hamesha ek side na ek entity (row, ex computer) other or bolo relation dusre ke ek-ek

Cardinality

One-to-one | one-to-many | Many-to-Many





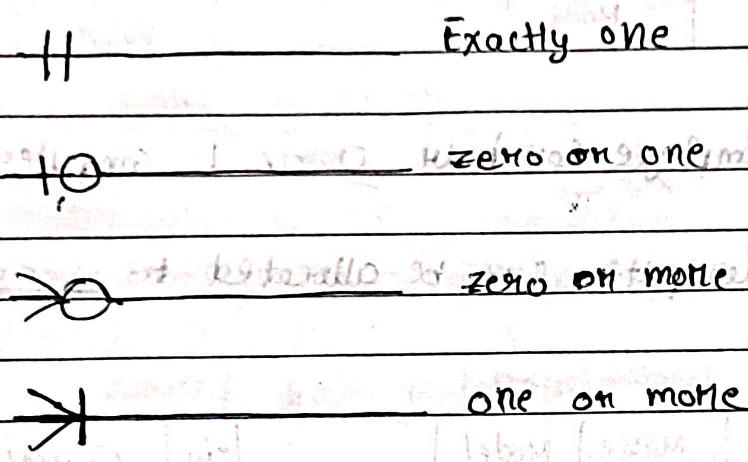
sate
than
udhar se ek
uthao
or
bolo
some
left
the
liye

1. 111
2 112
2 113

Date: / /

★ Crow-Foot Notation : is a visual representation used to depict the cardinality of relationship in an ER model

★ The notation comprises of 4 symbols and 1 of them needs to be used for representing each entity in a relationship



★ Computer can be allocated to one and only one emp.

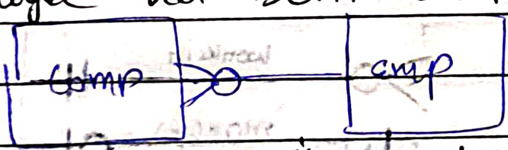
Ek computer sirf ek employee ko assign ho skta.

★ Employee can have zero or more computers.

Ek employee kol boht computers ho skde ne.

or cuts ve sare then udhar se ek uthao or bolo same left te liye

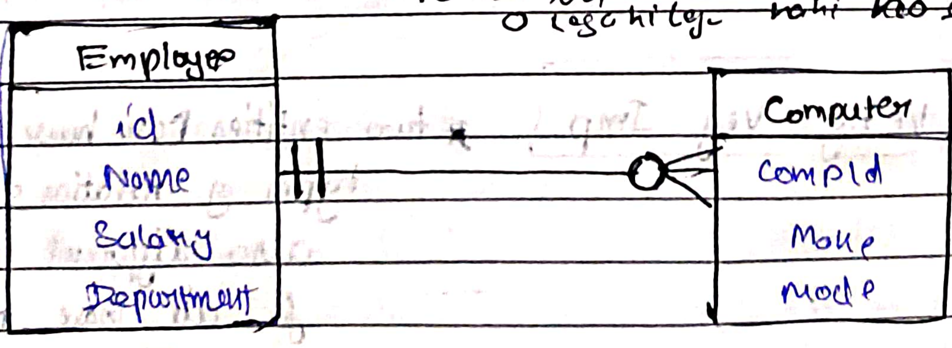
Ex.



if ek employe ne 2 leete and to koi tan 0 legehi ty.

Ek computer goi kise nu assign hgat tan kise hor nu ~~nahi kro skda~~

V. IMP



RAJDHANI

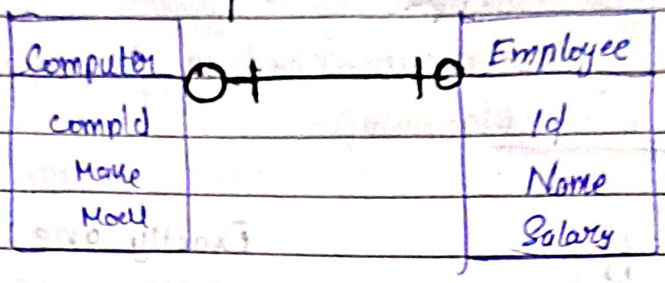
V.V. Imp

Date: / /

* Foriegn Keys must be created in tables in order to establish the relationship b/w the entities.

Example 1:

same as o+ ≡ +o



- Employee can have 0 or 1 computer
- Computer can be allocated to 0 or 1 Employee.

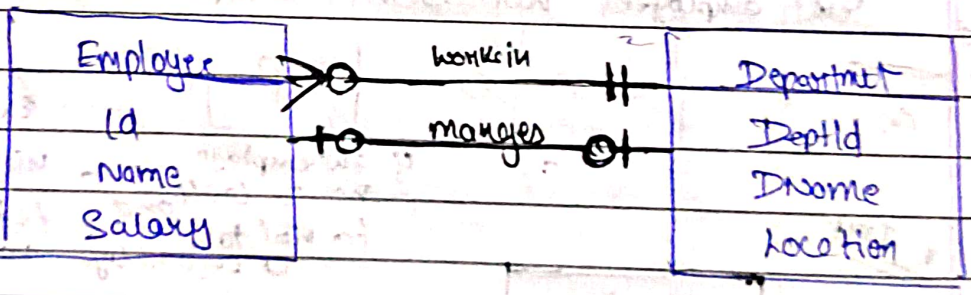
FOREIGN KEY

CompId	Make	Model
1001	Dell	Vostro
102	Dell	Precision

EId	EName	Salary	CompId
1	Jones	72000	1001
2	Ethan	3500	NULL

Example 2:

* There is always 1 manager



* Very-Very Imp

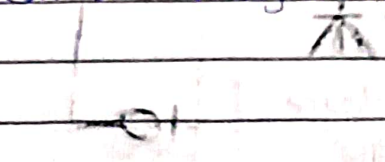
* two entities can have various types of relationships and also different cardinalities for all those relations.

Ex

* Employee can work in exactly one department.
* Ek department can have 0 or more employees

Date: / /

* ER Employee can manage 0 or 1 Dept.
* ER dept can be managed by 0 or 1 Emp.



* Foreign key.

• here DeptId will go to Employee table as m:1 many-to-one relation

for works in relationship

id	EName	Salary	*DeptId
1	James	72000	10
2	Ethan	85000	20
3	Emily	29000	10

(DeptId ← for works in relationship; foreign key)

(Foreign Key) references Employee Id

DeptId	DName	Locatio	*MgrId
10	ETA	Mysore	1
20	Ins	Banglore	NULL

* A (ek) department can have many Employees

★ 2 - Relationships ⟶ matlab ⟶ 2 - foriegn keys (Primary keys of table only)

V.V Imp

which one ⟵ depends on Cardinality ⟶ where to place

- one-to-one (on any side)
- one-to-many (on many side)
- many-to-many (Separate table containing both table's primary keys)

if you want to think why just take any example and think / this will start repeating in the primary key if you do vice versa of what said here!!

(and it's primary key is their combination.)

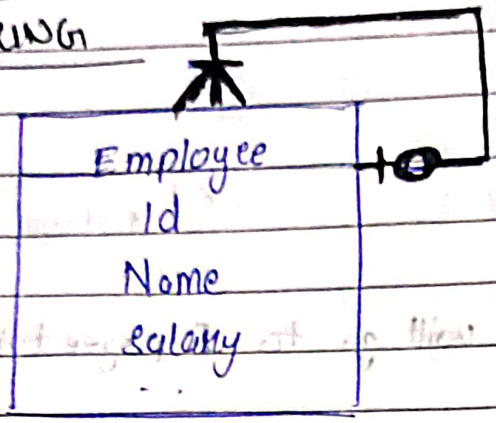
RAJDHANI

* Tricky Example

Date: / /

Example 3

SELF REFRENCING



* EN Employee reports to 0 or 1 manger.
* EN Manger(employee) can have many reportees (employees)

E.g.

* Foreign key in same Table only
*

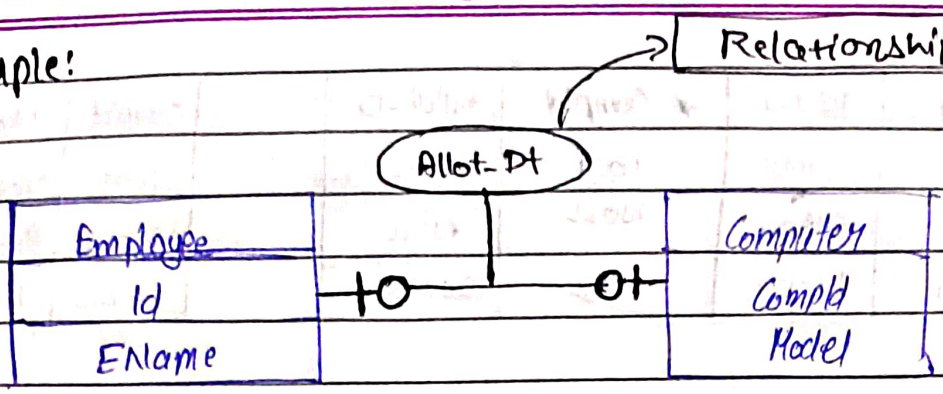
Id	EName	Salary	*MngrId → Foreign Key.
1	James	72000	1
2	Ethan	35000	Null

* The relation in which the foreign key will be created depends on Cardinality of the relationship

* one-to-one (1:1)
One is associated with a or one.
and or 0 will associate ← is one

Date: / /

example:



* in case of one-to-one (1:1)
  - Put the other's table Primary key and Allot_Dt to other other table as it's Foreign key. (either table).

* Allot_Dt - date on which employee was allocated a computer
  ↓
  It is neither property of employee table nor computer table.
  ↓
  It belongs to the relationship.
  so, represented differently.

* This relationship can be maped to Relations in either way:

V.V. Imp

1) use CompId attribute of computer table as Foreign key in the emp table.
(Allot_Dt along with the foreign key only)
(in the table where the foreign key is there)

2) Add Primary key of emp table EmpId as the Foreign key in the Computer table with Allot_Dt.

* Ek [illegible]ce can have multiple completers.

* Ek computer can be given to 0 or 1 employees.

* In 1:N relationship, the foreign key and relationship attributes are always added to (N) many side of the relationship

∴ the these attributes (primary key of Empid as foriegn key in Computer Table and the Allot_Dt also in Computer table along with the FOREIGN key.

RAJDHANI

The relationship is represented by a completely new table that has a composite Primary Key

⇓

{primary of 1, Primary of 2}

* That new table design requires
2- Foreign Keys on the new table linking the primary keys of each of the table.

* Attribute Also table goes with Foreign Key also

⇓

So it also go into new table.

Employee

Id	EName
1	James
2	Ethan
3	Emily

Computer

CompId	Model
1001	Testing
1002	;
1003	;

Id	CompId	Alot
1	1001	2014 -
3	1002	;
1	1003	;
3	1001	;

V. Imp

Try to read this

RAJDHANI

Date: / /

Summary:

- Entities, Relationships
  - ↓
  - * Define relationship
    - ↓
    - * Using foreign keys
  - * There can be multiple relationship between two tables.

* Types of Relationships

Date: / /

* Introduction:

Doubt: Single | and no ● → (?)

↓

Types of Database Systems:

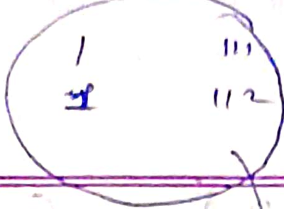
- Hierarchical (Trees)
  - each node can have only 1 parent
  - but multiple children (each can have)
- Network (Graph)
  - Each Node Connected to various
  - Multiple parents and children for each Node.
- Relational → one or more tables
  - Tables are related to each-other through parent child relationship
- NOSQL database (Not Only SQL) uses key-value pairs, graph, or document data structures to store data.
  - → These databases aim for simplicity of design,
  - → Horizontal scaling
  - → finer control over availability

Additional → OOP databases (DB4O) (ZopeDB)

Graph databases (Neo4J)

RAJDHANI

Q3



N.V. Imp

Date: / /

PRIMARY KEY decision

(one-to-many) (1 : N) — Koi ek → Many side vaali

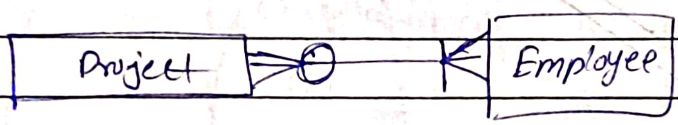
PRIMARY Key :

~~EmployeeId~~

Not Null (at least one)

(unique)

Q4



Many to Many

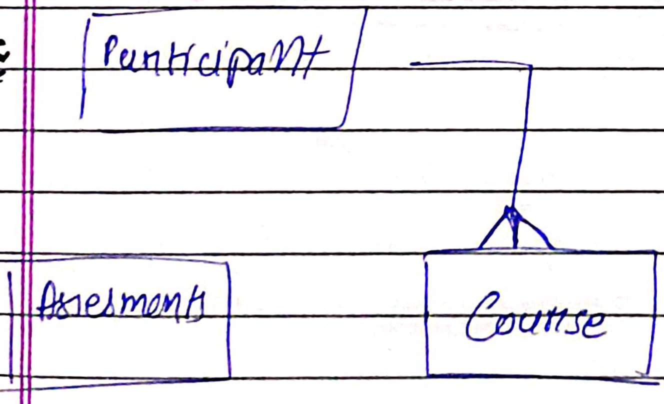
Both composite

Project	Employ
1	111
2	11
1	112

one-to-one

Koi bhi le lo. dono me se kisi ki bhi

Q5



DBMS - 02

Date: 26/02/2024

Normalization:

#1 Introduction: (Retail Shop example for data Redundancy, inconsistencies and Anomalies)

* Go see the case study

Summary * While designing database, relationship btw different attributes and related entities should be examined, to avoid
- → inconsistencies
- → Redundancy
- → Anomalies.

* **Functional Dependency:**

↓

"relation btw two attributes,

⟹ wherein one attribute can be uniquely determined from other attribute"

Ex: Roll No must have only one value associated with it in Name

* * In a relation R, two attributes/ set of Attributes A and B, B is functionally dependent on A. (A → B) if each value of A is associated with exactly 1 value in B"

* (Black Copy: Last ⟶) (Vishwadeep Gothi)

(Determinant) (dependent)

A → B

For the relation retailoutletstock (retailoutletId, itemcode, description, retailOutletLocation (street, city, zip), qtyAvail, retailunitprice, itemclass) *3

Several FDs are listed below:

{retailoutletid, itemcode} → qtyavailable	Closure of {retailoutletid, itemcode} gives all attributes of the class.
{retailoutletid, itemcode} → retailunitprice	
{retailoutletid} → retailoutletLocation	
retailunitprice → itemclass	
itemcode → description	

∴ {retailoutletid, itemcode} is the Candidate key

insertion

(null cannot be inserted)

Se hai.

Date: / /

★ Note: • Attributes part of the candidate key are called Prime attri
• Attributes that are not are Non-Prime attributes.

★ Full Functional Dependency:

Occurs when the dependent attribute is fully dependent on the whole Candidate Key and not a part of it.

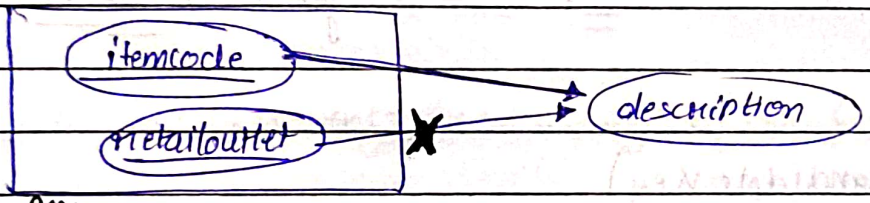
"Any subset of Candidate Key should not determine the dependent"

★2 Partial Functional Dependency:

In the retailoutletstock → description is unique for every ItemId, there is no need of retailoutletId

∴ description is Partially dependendent on {retailoutletId, itemcode}

attribute



"Any dependent attribute can be determined by a part of CK".

★3 Transitive Functional Dependency:

(non-key attribute)

V.V. Imp occurs when any dependent is indirectly determined by the Candidate Key

through non-key attributes

CK: A → B → C

A → C

A is CK

Go to last Page for Anomalies topic

★★ Full, Partial, Transitive Dependency kaise toh?

↳ First find the Candidate Key

↳ if part of candidate key can determine any attribute then Partial

↳ if full is required → Full

↳ if any type of transitivity is there b/w CK and the given FDs → Transitive

R (A, B, C, D, E, F)

Pehle Candidate key Nikalo

AB → CDEF

A → CDF

B → D

Full, Partial, Transitive dekhne ke liye.

{Candidate Key

↓

{AB} → like D or C are separately derivable from A or B

(Answer) (Partial)

Q4. Full (candidate key) se dekhte hai.

Q5 R(A, B, C, D)

Candidate key ki definition bhi toh Rahne dinge me

A → BCD

AD → BC

CD → AB

$A^+ = \{A, B, C, D\}$

$C^+ = \{C, A, B, D\}$

$AD^+ = \{A, B, C, D\}$

$CD^+ = \{A, B, C, D\}$

Imp

Q6- Business requirements that hold true for FD

Patientid, Vaccine → VaccineDate.

RAJDHANI

* Normalization : for reducing
→ inconsistency (either dyed fully or not at all)
→ Redundancy
→ Anomalies

Date: / /

"Decomposing larger tables into more organized smaller tables so as to minimize redundancy, inconsistencies and anomalies

* Normal forms : Set of rules based on which split in database is made.
↓
certain groups of attributes are analyzed.

1 NF
2 NF
3 NF

* 1 NF : (Koi relation functional Dependencies re nahi hoi is liye)
⇓ A relation R is said to be in 1NF if and only if :
1) All the attributes of R are atomic in nature.
2) There should be no multi-valued attribute.

Atomic Attribute : ⇒ simple and cannot be decomposed.

ex: in RetailStockOutlet → there is a composite attribute → Location
↳ zipcode, street, city

ex: If I want a result based on city → it will be difficult.
↳ so better if we have atomic attributes.

↳ Solution → Split the composite attribute into multiple atomic attributes.
↓ Result: table will be in 1NF

Multivalued Attribute : → like phone numbers

NULLS → Solution 1 → create separate columns (not good) ↓
Redundancy → Solution 2 ⇒ have multiple rows for each value of phone numbers (Primary key will repeat X)

Solution 3 → Create a new table with PKs of the table and Multivalued attribute values.

Data Lo
or
Redunda

we can't insert

we can't delete

Data Loss

Data Loss

RAJDHANI

Data Loss or Redundancy

CONSTRAINT naming PRIMARY KEY (Cordorld, Productid)
can be both table level + column level

Date: / /

* Issues with bad database design:

RetailOutletStock
CK → {RetailOutletId, ItemCode}

Ek hi table me sab mixed hai inhone

1. Insert Anamoly:

RetailOutletId	RetailOutletLocation	ItemCode	Description	Qty
~~R1001~~	~~Gear street,~~	~~NULL~~		

* If a new retail outlet is to be added, but it does not offer any items as of now → still we need to add

↳ ✗ this cannot be done because {retailoutletId, itemcode} is PRIMARY KEY here → ∴ itemcode cannot be NULL

"When some data in a relation cannot be added due to absence of other data"

2. Delete Anomaly:

RetailOutletId	RetailOutletLocation	ItemCode	Description	Qty
R1001	Mg street	I1001	---	---
~~R1002~~ ✗	~~Sale Street~~	~~I1005~~	~~Box~~	

* We want to drop the sale of item I1005,
If we try to delete it
* We will also end up deleting the corresponding details of RetailOutlet

"When deleting some data results in unexpected deletion of other required data."

3. Update Anamoly:

RetailOutlet

* there was an item belonging to class C pehle par Price badhane pe vo class B me chala gaya.
Lekin jo class C vaala item tha vo toh gaya nc.
So Data loss is there.

4 When modification of some data results in unexpected deletion or modification of other data.

RAJDHANI

Data loss
Redundancy.

galat data

galat time pe
kahin kuch or
kahin kuch or

Date: / /

Anomaly: is an inconsistency or Irregularity in data that arise due to Insert, update or delete operation.

||

Anomalies are the result of Poor Database Design or Redundancies"

* **Data Redundancy**: when same data is present multiple times.

||

Redundant data will require more storage and lead to potential inconsistency.

(kahin update hua gaya – kahi nahi hua

||

Inconsistent data.)

* INF → Advantages → queries effectively as it removes ambiguity.
* Limitation (need for 2NF) → There are Partial Functional Dependencies. Like description can be determined from itemcode only.

Date: / /

→ Hence we have redundant data in the table, which can be eliminated by having a separate table for item details

2# 2NF ★ (Agar Composite key hoi hi nahi toh 2NF dekhne ka (Partial FD) check karne ka koi tuk nhi)

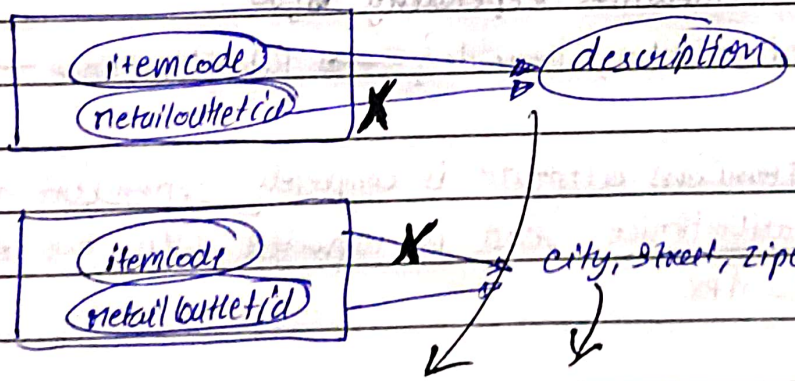
A relation R is in 2NF if and only if:

- R is already in INF.
- There is no-Partial dependency b/w non-key and key attributes.

* How to make a table 2NF?
  ↳ Remove all such Partial Dependencies and decompose the table.

Mantra ★ (Jitni Partial Dependencies Ohne tukde)

In RetailOutletStock, we have following partial dependencies: which must be removed (to make it 2NF)



Item (itemcode, description) RetailOutlet (retailoutletid, street, city, zipcode)

* Remaining table as it is with the composite primary key

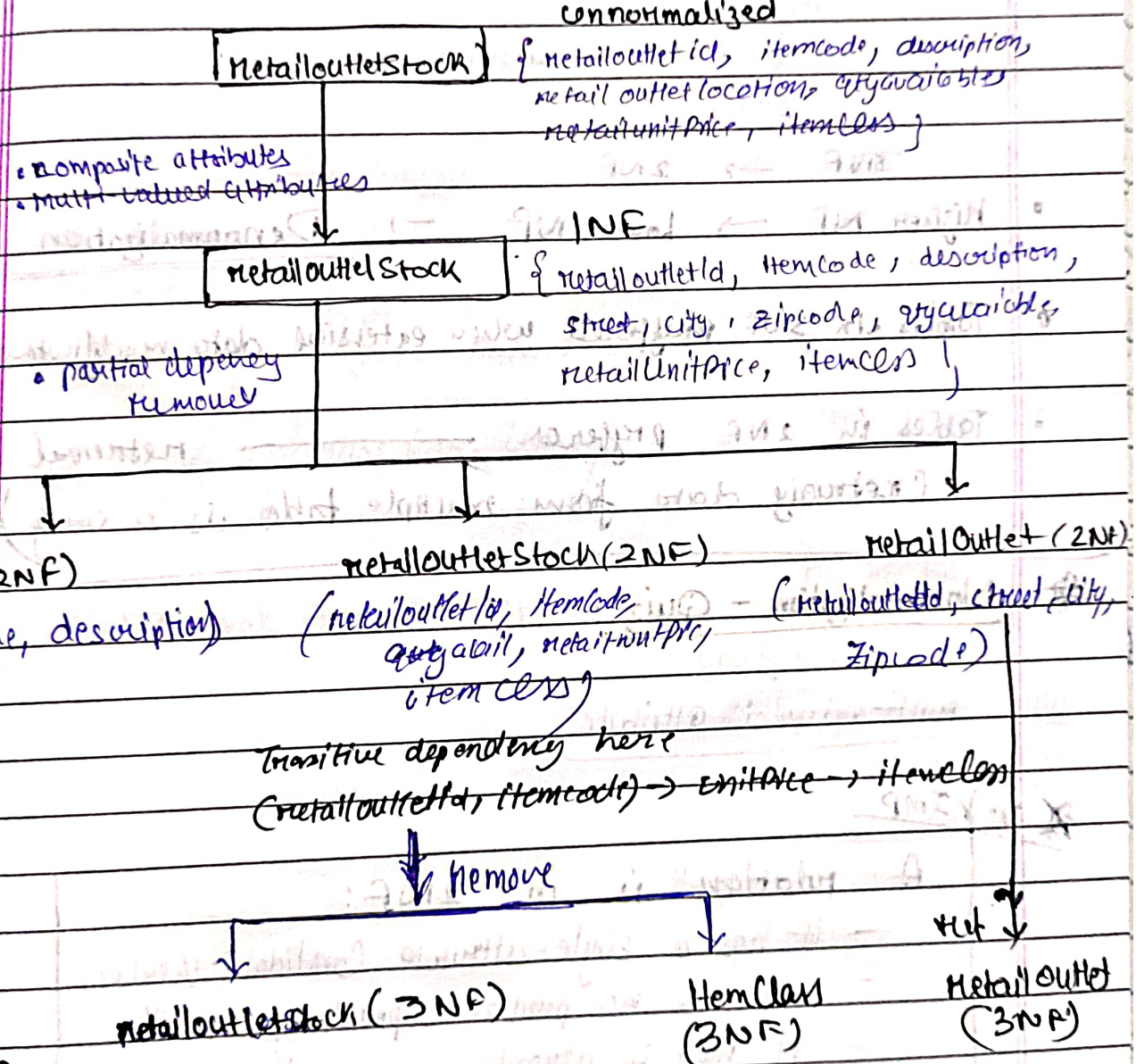
So tables after this:

* Now total tables Item, RetailOutletId, RetailOutletDetails, ItemClass, PhoneNumber Hajo nahi se Example dikha si.

Date: / /

* 3NF advantages: ensures data integrity, reduces the amount of data duplication

— Limitation → 3NF often leads to higher no. of smaller tables, can make queries more and more complex, may require combining multiple tables to retrieve data



RAJDHANI

Normalization - Quiz :

Q. Multi-valued Attribute

V. V. Imp

A relation is in 2NF:

- It has a single-attribute Candidate Key(s).
- There is no partial dependency.
- It must be already in 1NF

SQL Basics → DDL, DML, DCL, TCL

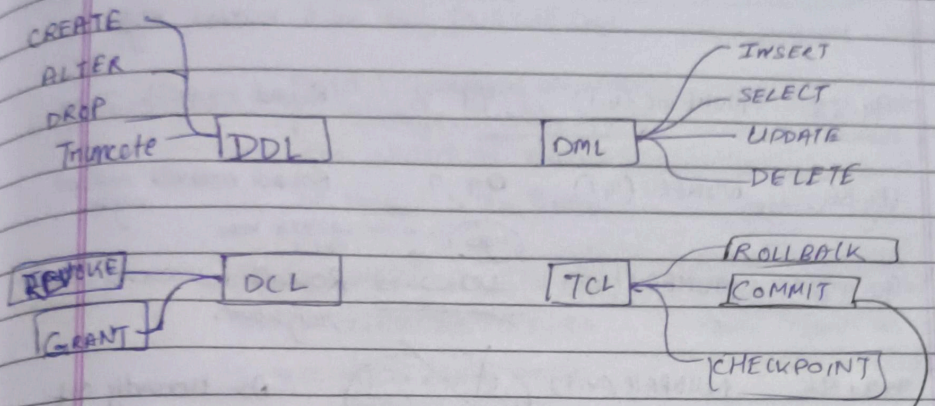
SQL (Structured Query Langage) 1970s

↳ High Level Language

↳ Platform independent and portable

↳ Standardized and fast.

Date: / /



Qus:

Q1. DCL TCL

Q2

Q5. To save the changes permanently in the database.

↓

TCL → COMMIT is used.

Q6. V.V. Imp

⇓

SQL DataTypes

#	Input	used Data Type	Stored As	Reason
1	134	INTEGER	134	Int ← decimal
2	234.6	INTEGER	234	Truncates when
3.	134	NUMERIC	134	Represented as it is.
4.	234.6	NUMERIC	235	Round towards nearest neighbor.
5.	134	NUMERIC(2)	Data Exception	Results in data loss so, it gives exception.
6.	23.6	NUMERIC(2)	24	Round towards nearest neighbor.
7.	23.5	NUMERIC(2)	23	if equidistant then Round towards
8.	23.4	NUMERIC(2)	23	↳ Nearest Neighbor.

RAJDHANI