The Rename Operation:

(i) First form $\rho_x(r)$

(ii) Second form $\boxed{A_1, A_2, \_\_\_ A_n}^{\,r}$ $\boxed{A_1 A_2 \_\_\_ A_n}^{\,x}$

* Final Output relation doesn't have any name.

Que:- Find out the id and name of the instructors who ~~are~~ earn more than instructor with id 12121

Instructor (Id, name, dept_name, salary)

$r_1 \leftarrow \Pi_{salary}(\sigma_{Id=1212}(\text{Instructor}))$

It is a relation not value.

So, First rename the ~~fir~~ Relation.

$(\rho_i(\text{Instructor}) \times \sigma_{Inst.id=12121}(\rho_w(\text{Instructor})))$

$\sigma_{i.salary > w.salary}$

$\Pi_{i.id,\, i.name}$

Query: Find the highest salary in the university.

$$\Pi_{salary}(Instructor) - \Pi_{r.salary}\left(\sigma_{r.salary < w.salary}\left(\rho_r(Instructor) \times \rho_w(Instructor)\right)\right)$$

All the salaries which are not highest

Additional Operations

→ Doesn't add any new power to relational algebra
(i) Set Intersection
(ii) Join
(iii) Division
(iv) Assignment

Division Operator: $r(R)$ and $s(S)$

To apply the division operation the condition is:
(i) $S \subset R$

→ The schema of $r \div s$ will be $R - S$.

r

A	B	C	D
1	5	2	7
1	5	3	7
1	6	3	7
2	6	2	7
2	6	3	7
3	6	2	7
3	5	3	7
3	5	3	7

s

C	D
2	7
3	7

$r \div s$

A	B
1	5
2	6
3	6

$$r \div s = \{t : t \in \Pi_{R-S}(r) \text{ and } \forall u \in S \text{ such that } t \times s \in r\}$$

Advanced relational algebra Operators:

(i) Generalized projection
(ii) Aggregation
(iii) Outer Join

} These operators

Generalized Projection:

$$\Pi_{F_1, \text{———}, F_K}(\vec{E})$$

A	B	C
1	1	2
1	5	7
7	9	10
1	5	20

$\Pi_{A, \cancel{A+}C}(r)$

A	B+C
1	3
1	12
7	19
1	25

Query Instructor (Emp_id, name, dept id, name, salary)
Display the employee ids along with salary + 20% of salary as ~~schem~~ bonus.

Ans: $\Pi_{Emp\_id,\ Salary \times 1.2}(\text{Instructor})$

(2) Aggregation: avg, mean, sum, multiply, count

numeric attributes

non-numeric attributes as well

$\underbrace{G_1, G_2, \text{———}, G_K}_{\text{optional}}$ $F_1(A_1), \text{———}, F_n(A_n)$

r

A	B	C
1	1	5
1	2	5
2	3	5
2	4	8
3	3	10

$G_{sum(C)}(r)$

sum(C)
23

$\rightarrow {}_{A}G_{sum(C)}(r) \rightarrow$

A	Sum(C)
1	10
2	13
3	10

Query: Instructor (Empl_id, name, dept_name, salary). Show the avg salary of every deparatment.

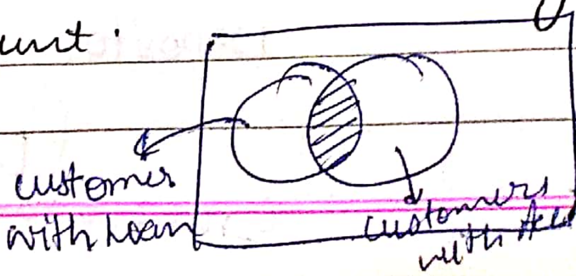
$${}_{dept\_name}G_{avg(salary)}(Instructor)$$

Que: # Banking Enterprise

Branch (b_name, b_city, assets)
~~Co~~ customer (c_name, b_city)
Account (A_no., b_name, balance)
loan (l_no, b_name, amount)
Depositor (c_name, Acc_no.)
borrower (c_name, Loan_no.)

Query: Find the names of all the customers having both a loan and an account.

Depositor and Borrower



A-(A-B)

$A \leftarrow \Pi_{c\_name}$ (Depositor)

$B \leftarrow \Pi_{c\_name}$ (Borrower)

✓ $\Pi_{c\_name}(\text{borrower}) \cap \Pi_{c\_name}(\text{depositor})$

X $\Pi_{c\_name}(\underbrace{\text{borrower} \cap \text{depositor}}_{\text{undefined}})$

wrong!!!

Query 2: Find the total ~~amount~~ ~~amount~~ balance in each branch.

Schema → Account

$_{bname}\mathcal{G}_{sum(balance)}$

Query 3: Find the average loan amount of each customer.

Aggregation

$_{customer\_name}\mathcal{G}_{Avg(amount)}(\text{borrower} \bowtie \text{loan})$

$_{customer\_name}\mathcal{G}_{avg.(amount)}(\sigma_{borrower.loan = loan.loan}(\text{borrower} \times \text{loan}))$

Query 4: Find all the customers who have an account in every branch ~~XYZ~~ ~~(branch city)~~ Jammu.

Depositor, Account, and Branch

$\Pi_{c\_name, b\_name}(\text{Depositor} \bowtie \text{Account}) \rightarrow$

$$\Pi_{b\_name}(\sigma_{b\_city='Jammu'}(Branch)) \longrightarrow$$

b_name

use division operators.

Query 5: Find the names of all depositors who have an account with a value greater than 6000.

$$\Pi_{C\_name}(\cancel{Scheme}\ \sigma_{balance>6000}(depositor \bowtie account))$$

Query 6: Find the names of all customers having a loan at "ABC" branch. → Branch name

$$\Pi_{C\_name}(\sigma_{B\_name=ABC}(loan \bowtie borrower))$$

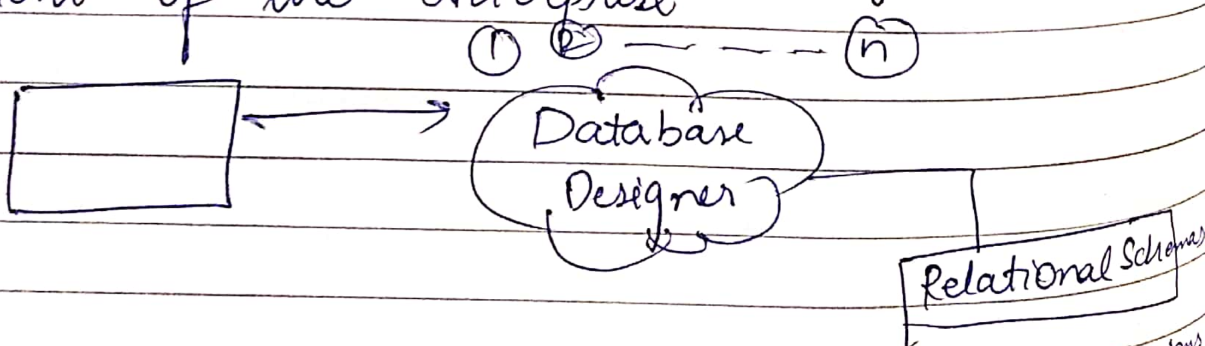
Module 2: Entity-Relationship Module

→ Design process:
→ Data Modelling
→ Modelling of the Constraints
→ Entity-Relationship Diagram
→ Design Issues
→ Weak Entity
→ Extended Entity-Relationship features.
→ Design of a bank database.
→ Reduction to Relational Schemas
→ Database Design.

(a) First principle: To understand the actual need of the end user.

(b) Second principle: Data model needs to represents the interprise into a set of relational schema

(c) Third principle: Understand the functional requirement of the enterprise



1. Logical Design → 2 different kinds of decissions
2. Physical Design

Logical Design

(i) Business decissions: what attributes one need to store.

(ii) Computer Science: How we can distribute the attributes accross the relational Schemas?

Physical Design:

Physical layout of the Database.

→ Data structures to be used for storage.

→ The block size for data transfer. & so on.

1. Enterprice Description: (Bank)

Organizational Description
→ The bank is organized into branches.
→ Each branch is located in a particular location, & is identified by a unique name
→ monitors its assets.

customer details
2. → Customers are identified by the C_id values.
→ Customer may take loan.
→ Customer is associated with a banker.

employees details
3. → Bank employees are identified by E_id.
→ Other information regarding the employees
→

Account Details
4. → Two types of accounts (a) Savings (b) Current.
→ Account operation details
→ Identified by unique account no.
→ balance, most recent date of access.
→ Savings account have the interest rate.

5. A loan can be held by one or more customers.

loan details
- → A loan is identified by unique L_no.
- → loan amount and loan payment.
- → A loan payment number does not uniquely identify a particular payment.
- → Date and payment

This is the input to designing the database.

- The final goal is to → come up with a set of relational schemas.

(1) We have to adopt a key design approach.
To use a data model for modelling the data.