Relational Model

Relation :-

A relation is a two dimensional table where columns are representing attributes of the relation and row representing tuples of the table.

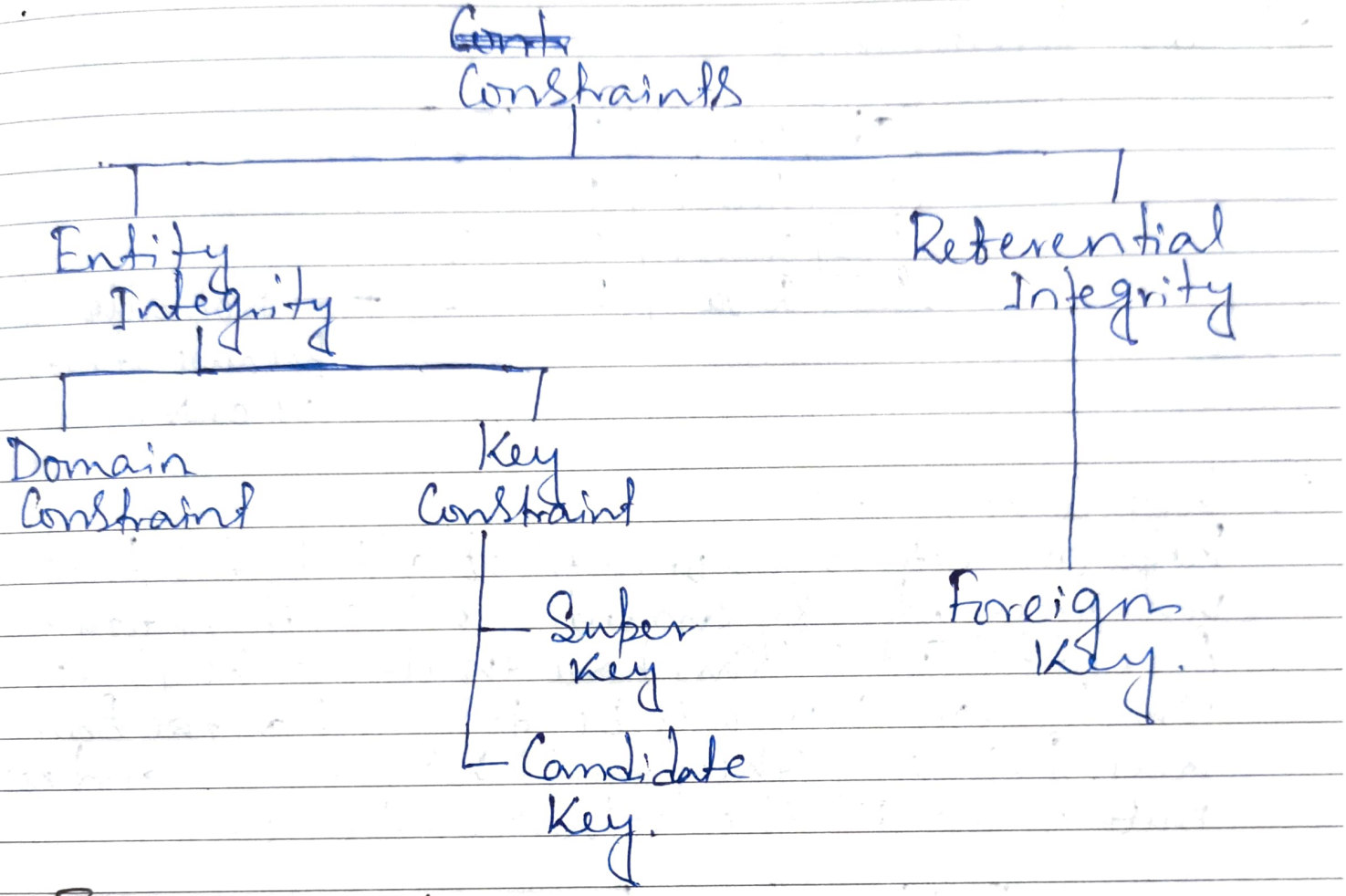
Course	Subject	Day
CS503	Algo	Monday
CS603	OS	Friday

Course - Subject

The name of the relation is :- Course - Suject
Attributes are :- Course, Subject, Day.
No. of tuples are :- 2
No. of tuples in a relation is called the Cardinality.
The Cardinality of the Course - Subject is 2. The tuples represents two different instances of the same relation.
When the value corresponding to an attribute at a particular instance is not known or not available, it is indicated with NULL value. Null is neither zero nor blank.
The relation with out tuples can be called as relation schema.

Constraints :-

A Constraint is a rule that restricts the values to be stored in the database.



Entity Integrity

The Constraint that restrict the values of a tuple in a relation. There are two types of Constraints.

1) Domain Constraint :-

Domain Constraint Specifies that the value of each attribute must be an atomic value from the domain of the respective attribute e.g. Say an attribute ~~Sex~~ married of a relation with domain {Y, N} and an attempt is made to enter 1 to mean Yes or 2 to mean NO then the system will indicate that the domain Constraint is violated and it should be Y for yes and N for no.

~~2) Key Constraint :-~~

Id	Name	Age	Married
8002	Ankit	23	N
8003	Sourav	25	Y
8005	Riya	22	1

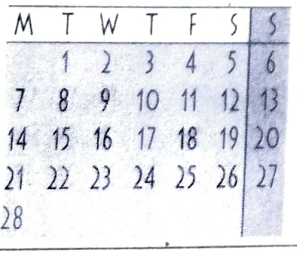
← Not allowed because it either Y or N.

2) Key Constraint :-
Keys are attributes or sets of attributes that uniquely identify an entity with in its entity set. An entity set E can have multiple keys out which one key will be designed as the Primary Key. Primary key must have unique and not null values in the relational table.

Id	Name	Age	Married
8002	Ankit	23	N
8003	Sourav	25	Y
8002	Tony	24	Y

↳ Not allowed as primary key values must be unique

Referential Integrity :-
Let us consider a situation when a supplier supplies certain product, which are being manufactured by certain company. Consider the relations :-



Supplier (id_no., name, item_code, manufactured_by)
Product (Product_code, Product_name, Company, manufacture_date).

The attribute item code in Supplier to match with Product_code in Product. Similarly the attribute Company of the relation Product should match with manufactured by of the relation Supplier. To refer to an attribute of one relation from another relation needs a link between the two relation.

Foreign Key :-

A foreign key is a set of attributes of one relation whose values are required to match values of some key attributes of another relations.

~~In the relation Supplier, the attributes item code~~

In other words, Foreign keys are the columns of a table that points to the Primary key of another table. They act as a Cross-reference between tables.

In the relation Supplier, the attribute item_code is the foreign key referring to the relation Product with attribute Product_code.

SUNDAY 27

id_no.	name	item_code	manufactured by
001	Ankit	186	Olive
002	Raj	172	Dove
003	Ankit	188	Clear

Supplier

Product_code	Product_name	Company	manufa date
186	Facewash	Olive	3.2.17
172	Soap	Dove	2.1.17
188	Oil	Clear	12.4.3

may contain another row.

Product

Single value Constraint:-
It refers that each attribute of any entity set has a single value. If the value of an attribute is missing in a tuple then we can fill it with a 'null' value.

SID	Name	class	Age	License no. Diving
801	Ankit	1st	19	DL-45629
802	Sourav	2nd	21	DL-43233, DL-32546
803	Ram	4th	20	

Allowed as a person mayor may not have a license.

Not allowed as a person does not have two driving licenses.

Relational Algebra:-

1. **Query Language:-**
A query language is a language in which user request information from the database. It can be categorized.

Procedural Lang:-
The user instructs the system to perform a sequence of operations on the database to compute the desired result.
Relational algebra is a procedural language.
~~It is this~~ Relational algebra operations can be divided into two broad categories:-
1. Set-oriented Operation.
2. Relation-oriented operation.

Nonprocedural lang:-

In this Lang. the user describes the information desired without giving a specific procedure for obtaining that information.
Example: Domain Relational Calculus.

Set-Oriented Basic Operations:-

Operations are:- Union, difference, intersection, ~~and Cartesian Product~~. The operations are Union compatible. Two relations $R(A_1, \ldots A_n)$ and $S(B_1, \ldots B_n)$ are said to be union compatible if they have same degree n and if $dom(A_i) = dom(B_i)$ for $1 \leq i \leq n$. This means that the two relations have the same no. of attributes and each corresponding pair of attributes has the same domain.

Union:-

The result of this operation, denoted by R ∪ S is a relation that includes all tuples that are either R or in S or in both R and S. Duplicate tuples are eliminated.

No.	Name
100	Smith
101	Ram
104	Hari
105	Ankit

R

No.	Name
100	Smith
102	Sham
103	Raju

S

No.	Name
100	Smith
101	Ram
102	Sham
103	Raju
104	Hari
105	Ankit

R ∪ S.

Intersection :-

The result of this operation denoted by $R \cap S$, is a relation that includes all tuples that are in both R and S.

No.	Name
100	Smith

$R \cap S$

Difference :-

The result of this operation, denoted by R-S, is relation that includes all tuples that are in R but not in S.

No.	Name
101	Ram
104	Hari
105	Ankit

R-S

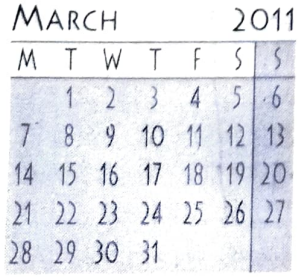
Cartesian Product in 10th March Thursday (*)

2. Relation-Oriented Operation :-

Relation-Oriented operation covers data manipulation through the Selection, projection, joins and division operations.

Selection :-

The Selection operation selects tuples from a relation that satisfy the given condition and the operation is symbolized by σ. The Syntax of the query is σ <conditions> (relation). The output of the query is also a relation. Logical Connectives and Comparison operators can be used in Selection Conditions.



The valid logical connectives are:
AND($\wedge$), OR($\vee$), NOT($\neq$)
Comparison operations are :-
$<, >, =, \geq, \leq, \neq$.

Example:-

Employee

Emp	Name	Dept.	Manager
100	Ankit	Produc.	Ajay
101	Raju	Account	Rana
102	Ram	Produc.	Ajay
103	Hari	Planning	Toppo
104	Sham	Account	Rana
105	Pooja	Security	Sachin

1. Find all the employees belong to Account dept from the relation employee.

Ans. $\sigma_{Dept.='Account'}(Employee)$

Emp	Name	Dept.	manager
101	Raju	Account	Rana
104	Sham	Account	Rana.

2. Find all employees working under the manager 'Ajay' from the relation employee.

Ans. $\sigma_{Manager='Ajay'}(Employee)$

3. Find the employees working in production and Planning dept.

Ans. $\sigma_{Dept.='product.' \wedge Dept.='planning'}(Employee)$
100, 102, 103

4. Find the employees working in 'Production dept. or the manager is Toppo.

Ans. $\sigma_{dept='Product' \vee Manager='Toppo'}(Employee)$
100, 102, 103.

5. Find the employees whose Emp is greater than 100 and less than 104 and working under the manager Rana.

Ans $\sigma_{Emp>100 \wedge Emp<104 \wedge Manager='Rana'}(Employee)$
101.

Projection Operator :-

The projection operation projects all or an attribute or a set of selective attributes from a relation. It is a vertical subset of the given relation. The operation is symbolized by Π. The syntax is $\Pi_{<list\ of\ attribute\ name/offset\ Positions>}(Relation)$.

Example :-

1. Show the Emp. and name of all employees in the relation employee.

Ans. $\Pi_{Emp.name}$ (Employee)
or
$\Pi_{1,2}$ (Employee)

100, 101, 102, 103, 104, 105.
Emp. Name only

2. Show the names of all the employees belongs to Account dept. from the relation employee.

Ans. Π_{name} ($\sigma_{Dept.='Account'}$ (Employee)).

Name
Raju
Sham.

3. Show the Emp. name and dept. working under the manager Ajay. from the relation employee.

Ans. $\Pi_{Emp.name, Dept}$ ($\sigma_{manager='Ajay'}$ (Employee)).

Emp. Name, Dept.
100
102

4. Show the employee number and manager name for those who are working in product. and planning.

Ans. $\Pi_{Emp, \text{manager}}$ ($\sigma_{Dept='product.' \wedge Dept='planning'}$ (Employee)).

5. Find the employee whose Emp is greater than 100 and less than 104, and working under manager Rama show all attributes.

Ans. $\Pi_{1,2,3,4}$ ($\sigma_{Emp>100 \wedge Emp<104 \wedge manager='Rama'}$ (Employee)).

APRIL 2011

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Join Operation :-

Join is a combination of a Cartesian Product followed by a Selection process. A join operation (combine) pairs two tuples from different relations, if and only if a given condition is satisfied.

Various types of join operations :-

Theta (θ) Join :-

The Conditional join is called θ-join.

Notation :- $R_1 \bowtie_\theta R_2$

Equi-Join :-

When Theta join uses only equality comparison operator it is said to be Equi-join

A

No.	Name
100	Smith
101	Rahul
102	Ankit
105	Raju
107	Robin
109	Sachin

B

No.	Name	Salary
100	Smith	5000
105	Raju	4000
107	Robin	3000

C

Emp_No.	Dept.
101	Account
107	Security
109	Planning

1. List of all employees those who have dept.

Ans. $\Pi_{Name}(\sigma_{Dept. \neq Null}(A \bowtie_{A.No = E.Emp\_No.} C))$

Name
Rahul
Robin
Sachin

=> Equi join.

Natural Join :-

Natural join does not use any Comparison Operator. It can only be performed if there exist is at least one common attribute exist between them. Those attributes must have same name and domain.

CID	Course	Dept.
CS01	Database	CS
ME01	Mechanics	ME

Course

Dept.	HOD
CS	Ajay
ME	Ram.

HOD

CID	Course	Dept.	HOD
CS01	Database	CS	Ajay
ME01	Mechanics	ME	Ram

Course ⋈ HOD.

Outer Join :-

All joint i.e. Theta join, Natural join, Equi join are called inner join. In Inner join includes only those tuples with matching attributes, rest are discarded in resulting relation. There exist methods by which all tuples of any relation are included in the resulting relation.

Three kinds :-

Left Outer join :- (A ⟕ B)

All tuples of Left relation A, are included in the resulting relation and if there exists tuples in A without any matching tuple in B then the B attributes of resulting relation are made NULL.

Course

A	B
100	Database
101	Mechanical

Left

HOD

C	D
100	Raj
102	Hari

Right

Course ⋈ HOD

A	B	C	D
100	Database	100	Raj
101	Mechanical	---	---

Right Outer Join (A ⟖ B):-

All tuples of right relation B are included in the resulting relation if there exists tuples in B without any matching tuple in A then A attributes of resulting relation are made NULL.

A	B	C	D
100	Database	100	Raj
---	---	102	Hari

Course ⟖ HOD

Full Outer join (A ⟗ B):-

All tuples of both Participating relations are included in the resulting relation and if there no matching tuples for both relations, their respective unmatched attributes are made NULL.

A	B	C	D
100	Database	100	Raj
101	Mechanical	---	---
---	---	102	Hari

Course ⟗ HOD

MARCH 2011

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

(#) Cartesian Product:-

No.	Name
100	Ram
101	Hari

A

Dept.
planning
Security

B

No.	Name	Dept.
100	Ram	planning
100	Ram	~~Security~~
101	Hari	~~planning~~
101	Hari	~~Security~~

A × B.

Division:-

The division operation is denoted by ÷. It is suited for query that include the phrase "for all". The relation A ÷ B is a relation on schema A-B contains all attributes of schema A that are not in schema B.

Example :-

P:

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_1
a_4	b_2
a_5	b_1
a_5	b_2

Q_1:

B
b_1
b_2

A
a_1
a_5

$P \div Q_1$

2.

name	Subject
A	DBMS
A	Java
B	DBMS
C	Java

Student

Subject
DBMS
Java

Subject_name.

Find the names of students who Passed in all the Subject

Student ÷ Subject_name.

Ans.

name
A

~~Ans.~~

Extended Relational Algebra Operation :-

1. Rename Operation :-

The results of relational-algebra are also relations but without any name. The rename operation allows us to rename that relation. 'Rename' Operation is denoted by

Small greek letter rho ρ.
Notation - $\rho_x(E)$
Where the result of Expression E is saved with name of x.

2. Aggregate functions:-

The type of queries that cannot be expressed in relational algebra is specified through aggregate functions. For example, to count the number of female employees and number of male employees the tuples are counted gender-wise.
The standard aggregate functions are :
→ Count → Sum → avg
→ min → max.

Syntax:-
$\mathcal{J}$ <function list> (R)
$\mathcal{J}$ → Script F.

EMP

Name	office	Dept.	Salary
Ram	400	CS	4500
Hari	200	Ecom	3500
Raju	100	Ecom	5000
Ankit	420	CS	6500
Abhi	500	Plan	6000

SUNDAY 13

1. Find the minimum Salary:-

$\mathcal{F}_{min(salary)}(EMP)$

min(Salary)
3500

2. Find the average Salary:-

$\mathcal{F}_{Avg(Salary)}(EMP)$

Avg(salary)
5100

3. Count the no. of employees in the CS dept:-

$F_{COUNT(name)}(\sigma_{Dept='CS'}(EMP))$.

COUNT(name)
2

4. Find the total Payroll for the Ecom dept.

$F_{SUM(salary)}(\sigma_{Dept.='Ecom'}(EMP))$.

Assignment Operation:-
The assignment operation is denoted by ←.

temp ← (E)

temp is the new name and E is the expression.

Relational Calculus:-
It is divided into two groups:-
1) Tuple relational Calculus:-

tuple variable (t_R) ranges for all tuple of relation
$t.A$ => Column A of tuple t.

Basic Form:- $\{t_1 A_{11}, t_2 A_{12}, t_3 A_{13} \ldots \mid Q\}$.

Calculus exp. ↑ Condition. one one many.

MARCH 2011
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Once the tuples satisfies the condition then ~~the~~ A11, A12 ... Column are taken out from the relation.

Example 1.1:- Student (Roll no., Name, Dept., Sex)
Find Roll no. and name of students in dept. 2.

Ans.
$\{t.\text{Roll no.}, t.\text{Name} \mid \text{Student}(t) \wedge t.\text{Dept} = 2\}$
↳ t tuple present in Student relation.

Example 2:- Find Roll no. and name of male student in dept. 2.

Ans.
$\{t.\text{Roll no.}, t.\text{Name} \mid \text{student}(t) \wedge t.\text{Dept.} = 2 \wedge t.\text{Sex} = \text{Male}\}$.

Example 2:- ~Depositor (cust_name, acc_no.)
~~borrow~~ borrower (cust_name, loan no.)
~loan (loan_no., branch_name, ~~amount~~ ammount)
~Customers (cust_name, city, Street)
Account (acc_no., branch_name, balance)
Branch (Branch_name, branch_city, assets)

2.1. Find the loan detail of loan above 1200.

$\{t \mid \text{loan}(t) \wedge t.\text{ammount} > 1200\}$
↳ bcz all attributes are required.

2.2:- Find the names of all customer who have a loan from branch 'x':

(from borrower and loan)

Ans. $\{ b.cust\_name \mid borrower(b) \wedge \exists l\,(loan(l) \wedge l.loan\_no = b.loan\_no \wedge l.branch\_name = 'x') \}$

$\Rightarrow l$ is bound variable.

⇓ OR

$\{ b.cust\_name \mid \exists b \in borrower \wedge \exists l \in loan \wedge l.loan\_no = b.loan\_no \wedge l.branch\_name = 'x' \}$

2.3:- Customer who have account or loan or both.

(We need to join customer and depositor, customer and borrower table)

$\{ t \mid customer(t) \wedge$ ___if borrower___ ①

or

___if depositor___ ②

① $\exists b(borrower(b) \wedge b.cust\_name = t.cust\_name)$

② $\exists d(depositor(d) \wedge d.cust\_name = t.cust\_name)$

$\{ t \mid customer(t) \wedge \exists b(borrower(b) \wedge b.cust\_name = t.cust\_name) \vee \exists d(depositor(d) \wedge d.cust\_name = t.cust\_name) \}$

MARCH 2011

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Example 3:- Student(<u>rollNo.</u>, name, degree, year, sex, deptNo., adviser)
dept.(<u>deptId</u>, name, hod, phone)
Course(<u>courseId</u>, cname, credits, dept.NO)
enrollment(<u>rollNO, courseId, Sem, year</u>, grade)

3.1 Determine the departments that do not have any girl students.

$\{ t.name \mid dept.(t) \wedge \neg (\exists S)(student(S) \wedge S.Sex = 'F' \wedge S.deptNo. = t.deptId) \}$

3-2 Obtain the names of courses enrolled by student named Mahesh.

~~$\{ S.name \mid Student(S) \}$~~

$\{ c.cname \mid Course(c) \wedge \exists e (C)(Course.Id = e.courseId) \wedge \exists s (e.rollNO = S.rollNO) \wedge S.name = 'Mahesh' \}$

$\{ c.Cname \mid Course(c) \wedge \exists e (enrollment(e) \wedge e.CourseId = c.courseId) \wedge \exists s (student(s) \wedge S.rollNO. = e.rollNO) \wedge S.name = 'Mahesh'. \}$

Quantifire → $\exists, \forall$
Bound variable → Quantified var
$\sum_{k=0}^{5}$ k changes 0 to 5
Free variable.

ii) Domain Relational Calculus:-

SQL depends on → tuple relation Calculus
QBE " " → Domain " "
(Query by example)

Domain variable → Ranges over domain of attributes.

Format:-

$$\{ x_1, x_2, x_3, \ldots \mid \text{cond}(x_1, x_2 \ldots) \}$$

$x_1, x_2, x_3, \ldots$: Domain variables.
cond. over domain variable.

Example 1:- Employee (FirstName, LastName, Eid, DOB, Add., Sex, Salary, Dno).
Dept. (Dno, Dname, Mid)

1.1 List the name and address of the employee whose name is Foo Bar.

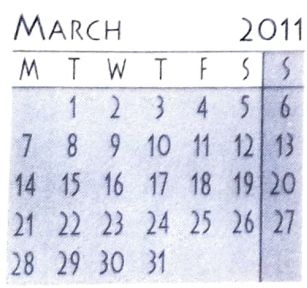
abcdefgh & xyz are domain variables.

Ans $\{abe \mid \exists c \exists d \exists f \exists g \exists h (\text{Employee}(abcdefgh) \wedge (a = \text{'Foo'}) \wedge (b = \text{'Bar'}))\}$ ①

OR

Domain var which is not present in (Emp ...) ① → ②
we can eliminate them.
they are optional so $\exists c \exists d \exists f \exists g \exists h$ are optional

$\{abe \mid \text{Employee}(abcdefgh) \wedge (a = \text{'Foo'}) \wedge (b = \text{'Bar'})\}$ ②.



OR ↓↓

$\{abe \mid Employee('Foo', 'Bar', c, d, e, f, g, h)\}$

12!- List the name of employee who have no department to manage?

We need to find the name so that the Eid in employee does not matched with Mid in dept.

Ans $\{ab \mid \exists c (Employee(abcdefgh) \wedge \neg \exists_2 (Dept.(xyz) \wedge z = c))\}$

↓ Domain of c

QBE :-

Query-by-example is ~~the name of~~ a database query lang. for relational databases. It is first graphical query lang. using visual tables where the user would enter commands. Today, some database systems for personal computers support variants of QBE lang.

QBE queries are expressed by example. Instead of giving a procedure for obtaining the desired answer, the user gives an example of what is desired. (SQL).

SUNDAY 20