

## Relational model

Originates from concept of relation

"Relation" is mathematical concept based on the ideas of sets

relation database stores data in the form of tables.

After designing the conceptual model of database using ER diagram  $\rightarrow$  relational model.

(RDBMS)

RDBMS lang: Oracle, SQL, MySQL etc.

Relational model  $\rightarrow$  table (row with columns)

$\Rightarrow$  Tuple  $\rightarrow$  row

Each table of the column has a name or attributes

primary key (Students) Attributes/ Column

Roll No.	Name	Phone No.
1	Ajay	989857282
2	Ray	9899972841
3	Vijay	8870887627
4	Aman	8067199167

Domain (09)

Range  $\rightarrow$ , Cardinality = 4

Relation Student (Roll-as primary key)  
Phone no.

- \* Relation → Table of ~~Content~~.
- \* Attribute → column.
- \* Domain :- set of allowable values for one or more attributes.
- \* Tuple :- row.
- \* Relational schema :- represent name of relation with its attributes.
- \* Relation instance :- finite set of tuples never have duplicate value.
- \* Degree :- total no. of columns in table.
- \* Cardinality :- no. of rows in the table.
- \* Relation key :- Every row has one or multiple attributes, that can uniquely identify the row in the relation, which is called relation key (Primary key).

- \* Tuple variable :- data stored in a record of the table.

### → Properties of Relation Model ↴

- \* Each relation has a unique name.
- \* Each row is unique. No duplicate row.
- \* Entry in any column have the same domain.
- \* Each column has a unique name.
- \* Order of ~~new~~ column or rows is irrelevant.
- \* Each cell of relation contains exactly one value i.e. attribute value are required to be atomic.

## alternative Terminology

formal terms	alter 1	alternatives
Relation	Table	file
Tuple	Row	record
Attribute	Column	field

Integrity Constraints :- It is used to guard the db against accidental damage to the db.

- \* set of rules that db is not permitted to violate.
- \* it ensure that changes (update, deletion, insertion) made to the db by authorized user do not

### Type

- ① Domain Constraints
- ② Entity Integrity
- ③ Referential Integrity
- ④ Eg. Constraints

\* Domain Constraints → defines the domain on the valid set of values for an attribute.

\* data type of domain → string, char, int, time, date, currency etc.

The value of attribute must be available in the corresponding domain.

PAGE NO. \_\_\_\_\_

DATE: \_\_\_\_\_

- \* Entity integrity constraint  $\rightarrow$  primary key value should not be null.

$\Rightarrow$  Because primary value is used to identify individual rows in relations and if the primary key has a null value, then we can't identify these rows.

- \* Referential Integrity constraints :- b/c two tables

when foreign key references the primary key of a table.

- \* Referential integrity constraints  
every value of primary key should be foreign key or null.

attribute  
PK  
foreign  
key

- \* You can't delete primary key record if it is in foreign table.
- \* You can't change primary key value in the primary table.

- \* You can't insert value outside primary key value in the foreign key.

\* foreign can be null.

## Key constraints

- \* candidate keys , out of one then those will be primary key
- \* Primary value should be unique.
- \* value of primary key must not be null

## SQL

SQL data type → specify type of data

Each column variable has a related datatype in SQL  
→ used for creating table and according to key.

- Numeric → bigint, int, smallint, tinyint, bit, decimal, numeric
- String → char, varchar, ~~or~~ varchar(max), Text
- Date and time → date, time, timestamp, interval ✓  
(date + time)
- Interval : ~~or~~ per
- \* Binary → Max length of 8000 bytes (fixed-length binary data)
- \* Vartbin (Binary, Varbinary, varbinary(max))

## Miscellaneous Data Types

\* ~~Sql~~ SQL-Variant, Timestamp, UniqueIdentifier, XML, cursor, Table

↳ stores values of various SQL Server supported data types

PAGE NO. 12

Reference to a cursor object stores a set of rows in memory, which gets updated when rows are updated.

set of rows for later processing.

Author: Gurbaj Singh

unique identifier

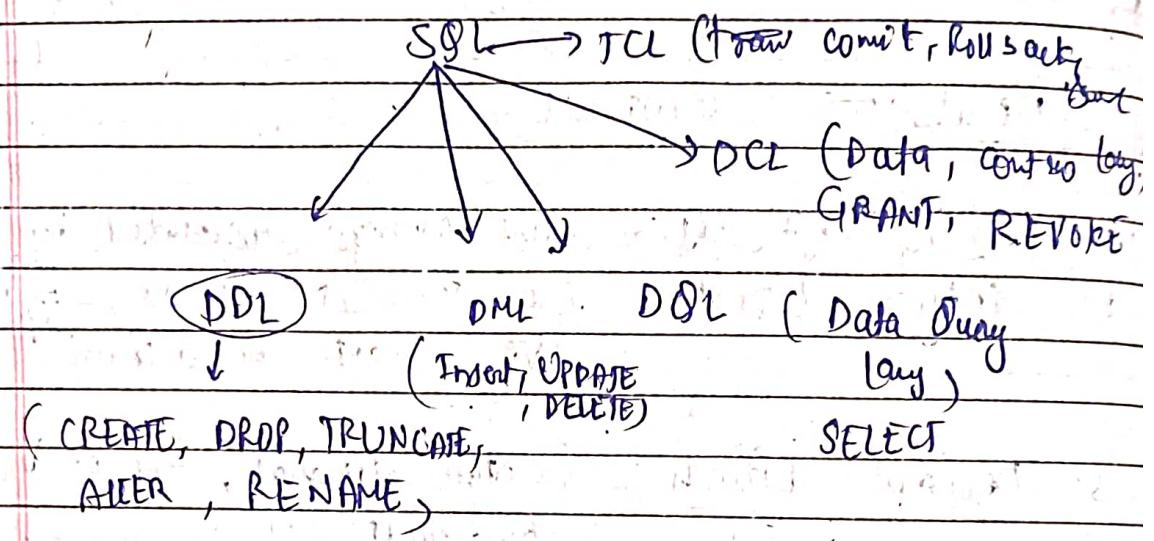
## Literals → Constant

1. String Literals: are always surrounded by ()
2. Integer Literals: can be +ve or -ve, by default +ve can't be decimal
3. Decimal Literals: contain decimal, +ve or -ve by default +ve
4. Datetime Literals: for datetime values

What

## Q. What is SQL?

Ans "structured Query language", every relational db db software, interact with a language known as SQL lang because it's a simple english like lang which guidelines are provided by adopted by standard Organisation  $\rightarrow$  ANSI adopted by all database vendors, like oracle, mySQL, microsoft etc.



### Characteristics of SQL

- \* SQL is easy to learn.
- \* used to access data from RDBMS.
- \* SQL can execute queries against db.
- \* SQL used to describe data.
- \* SQL used to define data in db and manipulate it when needed.
- \* SQL is used to create and drop db and file.
- \* SQL allows users to set permission on tables, procedures, views.

## Advantages of SQL

- \* High Speed → using SQL query, user can quickly & efficiently retrieve a large amount of data from db.
- \* No coding needed :-
- \* Portability :- SQL can be used in laptops, PCs, server and phone.
- \* Interactive language :- SQL is a domain lang. (DBMS) communicate with db, retrieve answers in sec. from db.
- \* Multiple data view :- Using the SQL lang, the users can make different views of the db.

## Operators

- ① Unary operator ( +, -, !, ++, -- )
- ② Binary operator ( + arithmetic, concatenation, relational, logical, another relation operator )
- ③ Set operator ( Union, union all, Minus, Intersect )

## vi views in Database

- Virtual table
- View is the result set of stored query  
we can't update data on basic table from view.
- Read-only vs. Updatable views → data can be
- Materialized views.

[Create view V, as select id - from student.]  
virtual doesn't exist actually but above  
query compile only.

## Advantage of view $\Rightarrow$

- \* To restrict data access
- \* To make complex query easy
- \* To provide data independence
- \* To present diff views of the same data

## Index In SQL

↓      ↓  
used to retrieve      User      used to speedup.  
data from DB      can't see      search  
quickly      indexes.

Grant CREATE INDEX idx-lastname  
on Employees (Lastname);

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

mapping  
Query and sub Query

PAGE NO. \_\_\_\_\_  
 DATE: \_\_\_\_\_

Eid	E-name	Dept	salary
1	Ram	HR	10000
2	Amrit	MRKT	20000
3	Ravi	HR	30000
4	Nitin	MRKT	40000
5	Varan	IT	50000

Ques: find max. from 'Salary' using  
 SPL: query

SPL

Select Max (Salary) from Emp  
50000

Ques: display name of emp who is taking max.  
 salary.

Ans

Select Max (Salary) from Emp  
 → outer query

Select E-name from

Emp where Salary = select Max (Salary)

Inner query

execute 1st

Ques: write the SPL Query to display 2nd highest  
 salary from Emp table.

Ans: Select (Salary) from  
 Emp where salary < (Salary - Max (Salary) from Emp);

50000

{  
 10000 ✓  
 20000 ✓  
 30000 ✓  
 40000 ✓  
 50000 X

Select Max(salary) from

Emp where salary > (sal select as Max(salary) from Emp)

40000

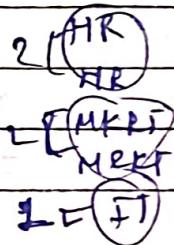
② Select E-name from Emp.

where salary > (select max(salary) from Emp)

NITIN

Ques 4: Write query to display all the names of dept names along with no. of employees working in that

HR	2
MKT	2
IT	1



Group by function

from Emp,

Select ~~from Emp~~ <sup>dept, count(dept)</sup> from Emp group by dept.

Ques Write a query to display all dept names where no. of deps are less than 2.

Ans

Select dept, ~~from Emp~~ group by dept  
having count(dept) < 2

note: we are having in groupby instead of where

② find name add of Emp. of also

PAGE NO.

DATE:

Select

Select E-name from emp where dept in

where dept in select dept from emp group by dept having co

Ques

\* Create Emp.

\* Insert data in an table

\* Alter by adding new column

\* Deleting rows from the table

\* Alter table by change data type of Emp. Id

## Create Table Command

```

Create table <table-name> (column1 name datatype,
                           column2 name datatype,
                           column3 name datatype,
                           ...);
disc table-name;

```

createTable Emp  
 (id int,  
 name varchar(20),  
 salary number(10))  
 );  
 disc Emp;

use to describe

detail of table

## Alter Command

→ Add column

→ Remove "

→ Modify datatype;

→ Modify " length

→ add constraint

→ Remove constraints

→ Rename column / table

Student

.int:	ID	name	varchar(20)
	-	-	-

\* Alter table student Add (address varchar(20));  
 → new col. address will be added

\* Alter table ~~Emp~~ student drop column address;  
 column address will be deleted

\* Alter table student modify id varchar(10);  
 datatype modified

\* Alter table student rename column id to rollno;  
 (name of col. will change from id → rollno)

\* alter table student xname to Emp;

false name → Emp

\* Alter table Emp add primary key (rollno)

Alter	Update
• DDL	• DML
• Add, delete column	changes only data <code>set salary = salary * 2</code>
• change datatype of datatype	<code>update Emp</code> <code>set salary = salary * 2</code>
• ID → Eid	<code>update Emp</code> <code>set</code>
• change table name	

Delete	Drop	Truncate	Truncate table
• DML	• DDL	• DDL	• DDL
Delete from <u>Student</u> , delete rows	D		
all rows		Truncates entire table	
deleted in upper query			