1) One to one
2) One to many
3) Many to one
4) Many to many

Lecture-9
DC
01/02/2018

SQL

Storename	Sales	Date
Kolkata	1500	05-Jan-99
Bombay	800	08-Jan-99
Kolkata	200	07-Jan-99
Delhi	700	05-Jan-99
Chennai	700	06-Jan-99

SQL> select distinct storename from store;

Select associates with column
where associates with row

For ANSI SQL
like for strings
= for numerics
<> for !=

SQL> select * from store where storename like 'Bombay'
where sales = 700

SQL> select storename from store where sales > 1000 or
(sales < 500 and sales > 275);
→ key needs to be specified again for comparison.

SQL> select * from store where storename in ('Kolkata', 'Bombay');

Operator precedence is same as ANSI C.

SQL> select * from store where date between '6-Jan-99' and '9-Jan-99';

SQL> select * from store order by sales;

↳ default is ascending

order by sales asc;] explicitly specifying
order by sales desc;]

SQL> select * from store order by (sales, date);

first sort by date and then by sales for items with same date.

Sales

snum	sname	city	comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelrod	New York	0.10

customer

cnum	cname	city	rating	snum
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass Clemens	Berlin	300	1002
2005	Clemens	London	100	1001
2006	Pereira	San Jose	300	1007
2007	Pereira	Rome	100	1004

Orders

Onum	Amt	Odate	Cnum	Snum
3001	18-69	10/03/1990	2008	1007
3003	767.19	"	2001	1001
3002	1900.10	"	2007	1004
3005	5160.45	"	2003	1002
3006	1098.16	"	2008	1007
3009	1713.23	10/4/90	2002	1003
3008	75.75	"	2004	1002

Inbuilt functions

Aggregate func^n produces a single value for entire group of table entries. There are 5 -

COUNT() →
MIN()
MAX()
SUM()
AVG()

SQL> select SUM(Amt) from sales;
SQL> select MIN(Amt) from sales;

SQL> select COUNT(*) from sales; → counts null values as well

SQL> select distinct COUNT(A) from sales; → counts distinct values.

SQL> select amt + balance from sales; → single column with sum.

amt + balance
⋮

SQL> select amt + balance as "ABC" from sales;

ABC
⋮

SQL> select max(amt + balance) from sales; → func^n on calc^n.

Group by

This clause allows defining a subset of the values in a particular field in terms of another field and apply an aggregate func^n to the subset. This enables combining fields and aggregate func^n ~~and fields~~ in a single select statement

Q. Find largest order taken by each sales person.

SQL> select snum, max(amount) from orders group by snum;

shouldn't be p. key

↳ not a primary key.

operation is right to left. First grouping & then operation.

Snum	Sname	amt.	Rollno. → primary key
1001	Hemant	500	1
1001	Shreya	200	2
1001	Arnab	50	3

group by

Output :

Snum	Max(amt)
1001	9891.88
1002	5160.45
1003	1713.23
1004	1900.10
1007	1098.16

SQL> select roll, name, max(tot marks) from student group by dept ;

Q. Find the largest order taken by each salesperson on each date.

Lecture-8
SSK