CREATED BY: SACHIN BHARDWAJ, PGT(CS) KV NO.1 TEZPUR & VINOD KUMAR VERMA , PGT(CS) KV OEF KANPUR

Database Query Using SQL

Lets do practical on DATABASE…

SORTING OUTPUT

By default records will come in the output in the same order in which it was entered. To see the output rows in sorted or arranged in ascending or descending order SQL provide **ORDER BY** clause. By default output will be ascending order(ASC) to see output in descending order we use DESC clause with ORDER BY.

Select * from emp order by name; (ascending order)

Select * from emp order by salary desc;

Select * from emp order by dept asc, salary desc;

AGGREGATE functions

Aggregate function is used to perform calculation on group of rows and return the calculated summary like sum of salary, average of salary etc.

Available aggregate functions are –

1. SUM()
2. AVG()
3. COUNT()
4. MAX()
5. MIN()
6. COUNT(*)

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000

Select SUM(salary) from emp;

Output – 161000

Select SUM(salary) from emp where dept='sales';

Output - 59000

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000

Select AVG(salary) from emp;

Output – 32200

Select AVG(salary) from emp where dept='sales';

Output - 29500

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000

Select COUNT(name) from emp;

Output – 5

Select COUNT(salary) from emp where dept='HR';

Output - 1

Select COUNT(DISTINCT dept) from emp;

Output - 3

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000

Select MAX(Salary) from emp;

Output – 45000

Select MAX(salary) from emp where dept='Sales';

Output - 35000

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000

Select MIN(Salary) from emp;

Output – 24000

Select MIN(salary) from emp where dept='IT';

Output - 27000

AGGREGATE functions

Empno	Name	Dept	Salary
1	Ravi	Sales	24000
2	Sunny	Sales	35000
3	Shobit	IT	30000
4	Vikram	IT	27000
5	nitin	HR	45000
6	Krish	HR	

Select COUNT(*) from emp;

Output – 6

Select COUNT(salary) from emp;

Output - 5

count(*) Vs count()

Count(*) function is used to count the number of rows in query output whereas count() is used to count values present in any column excluding NULL values.

Note:

All aggregate function ignores the NULL values.

GROUP BY

GROUP BY clause is used to divide the table into logical groups and we can perform aggregate functions in those groups. In this case aggregate function will return output for each group. For example if we want sum of salary of each department we have to divide table records.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Aggregate functions by default takes the entire table as a single group that's why we are getting the sum(), avg(), etc output for the entire table. Now suppose organization wants the sum() of all the job separately, or wants to find the average salary of every job. In this case we have to logically divide our table into groups based on job, so that every group will be passed to aggregate function for calculation and aggregate function will return the result for every group.

Group by clause helps up to divide the table into logical groups based on any column value. In those logically divided records we can apply aggregate functions. For. E.g.

```
SELECT SUM(SAL) FROM EMP GROUP BY DEPT;
SELECT JOB,SUM(SAL) FROM EMP GROUP BY
DEPT;
SELECT JOB,SUM(SAL),AVG(SAL),MAX(SAL),COUNT(*) EMPLOYEE_COUNT FROM EMP;
```

NOTE :- when we are using GROUP BY we can use only aggregate function and the column on which we are grouping in the SELECT list because they will form a group other than any column will gives you an error because they will be not the part of the group.
For e.g.

```
SELECT ENAME,JOB,SUM(SAL) FROM EMP GROUP BY JOB;
```

Error -> because Ename is not a group expression

HAVING with GROUP BY

- If we want to filter or restrict some rows from the output produced by GROUP BY then we use HAVING clause. It is used to put condition of group of rows. With having clause we can use aggregate functions also.
- WHERE is used before the GROUP BY. With WHERE we cannot use aggregate function.
- E.g.
- **SELECT DEPT,AVG(SAL) FROM EMP GROUP BY DEPT HAVING JOB IN ('HR','SALES')**
- **SELECT DEPT,MAX(SAL),MIN(SAL),COUNT(*) FROM EMP GROUP BY DEPT HAVING COUNT(*)>2**
- **SELECT DEPT,MAX(SAL),MIN(SAL) FROM EMP WHERE SAL>=2000 GROUP BY DEPT HAVING DEPT IN('IT','HR')**

CREATED BY: SACHIN BHARDWAJ, PGT(CS) KV NO.1 TEZPUR & VINOD KUMAR VERMA , PGT(CS) KV OEF KANPUR

MYSQL FUNCTIONS

A function is built – in code for specific purpose that takes value and returns a single value. Values passed to functions are known as arguments/parameters.

There are various categories of function in MySQL:-

1) String Function
2) Mathematical function
3) Date and time function

String Function

Function	Description	Example
CHAR()	Return character for given ASCII Code	Select Char(65); Output- A
CONCAT()	Return concatenated string	Select concat(name, ' works in ', dept,' department '); Output- name works in dept department
LOWER()/ LCASE()	Return string in small letters	Select lower('INDIA'); Output- india Select lower(name) from emp;
SUBSTRING(S, P,N) / MID(S,P,N)	Return N character of string S, beginning from P	Select SUBSTRING('LAPTOP',3,3); Output – PTO Select SUBSTR('COMPUTER',4,3); Output – PUT
UPPER()/ UCASE()	Return string in capital letters	Select Upper('india'); Output- INDIA
LTRIM()	Removes leading space	Select LTRIM(' Apple'); Output- 'Apple'
RTRIM	Remove trailing space	Select RTRIM('Apple '); Output- 'Apple'

String Function

Function	Description	Example
TRIM()	Remove spaces from beginning and ending	Select TRIM(‘ Apple ‘); Output-’Apple’ Select * from emp where trim(name) = ‘Suyash’;
INSTR()	It search one string in another string and returns position, if not found 0	Select INSTR(‘COMPUTER’,’PUT’); Output-4 Select INSTR(‘PYTHON’,’C++’); Output – 0
LENGTH()	Returns number of character in string	Select length(‘python’); Output- 7 Select name, length(name) from emp
LEFT(S,N)	Return N characters of S from beginning	Select LEFT(‘KV NO1 TEZPUR’,2); Output- KV
RIGHT(S,N)	Return N characters of S from ending	Select RIGHT(‘KV NO1 ’,3); Output- NO1

Numeric Function

Function	Description	Example
MOD(M,N)	Return remainder M/N	Select MOD(11,5); Output- 1
POWER(B,P)	Return B to power P	Select POWER(2,5); Output-32
ROUND(N,D)	Return number rounded to D place after decimal	Select ROUND(11.589,2); Output- 11.59 Select ROUND(12.999,2); Output- 13.00 Select ROUND(267.478,-2) OUTPUT- 300
SIGN(N)	Return -1 for –ve number 1 for +ve number	Select sign(-10) Output : -1 Select sign(10); Output : 1
SQRT(N)	Returns square root of N	Select SQRT(144); Output: 12
TRUNCATE(M, N)	Return number upto N place after decimal without rounding it	Select Truncate(15.789,2); Output: 15.79

Date and Time Function

Function	Description	Example
CURDATE()/ CURRENT_DATE()/ CURRENT_DATE	Return the current date	Select curdate(); Select current_date();
DATE()	Return date part from date-time expression	Select date('2018-08-15 12:30'); Output: 2018-08-15
MONTH()	Return month from date	Select month('2018-08-15'); Output: 08
YEAR()	Return year from date	Select year('2018-08-15'); Output: 2018
DAYNAME()	Return weekday name	Select dayname('2018-12-04'); Output: Tuesday
DAYOFMONTH()	Return value from 1-31	Select dayofmonth('2018-08-15') Output: 15
DAYOFWEEK()	Return weekday index, for Sunday-1, Monday-2, ..	Select dayofweek('2018-12-04'); Output: 3
DAYOFYEAR()	Return value from 1-366	Select dayofyear('2018-02-10') Output: 41

Date and Time Function

Function	Description	Example
NOW()	Return both current date and time at which the function executes	Select now();
SYSDATE()	Return both current date and time	Select sysdate()

Difference Between NOW() and SYSDATE() :

NOW() function return the date and time at which function was executed even if we execute multiple NOW() function with select. whereas SYSDATE() will always return date and time at which each SYDATE() function started execution. For example.

mysql> Select now(), sleep(2), now();

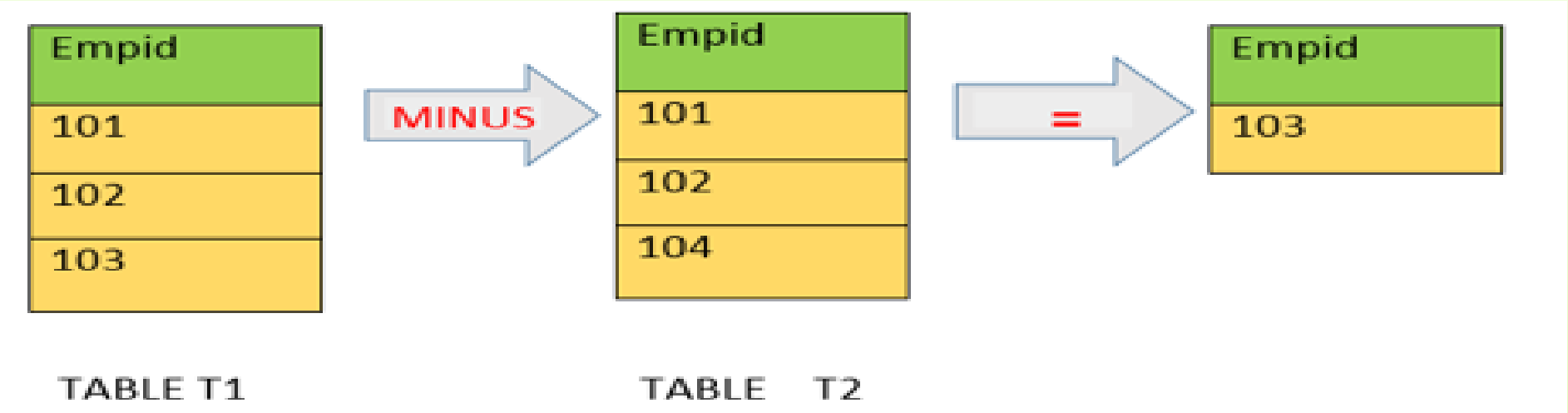
Output: 2018-12-04 10:26:20, 0, 2018-12-04 10:26:20

mysql> Select sysdate(), sleep(2), sysdate();

Output: 2018-12-04 10:27:08, 0, 2018-12-04 10:27:10

SQL – MINUS

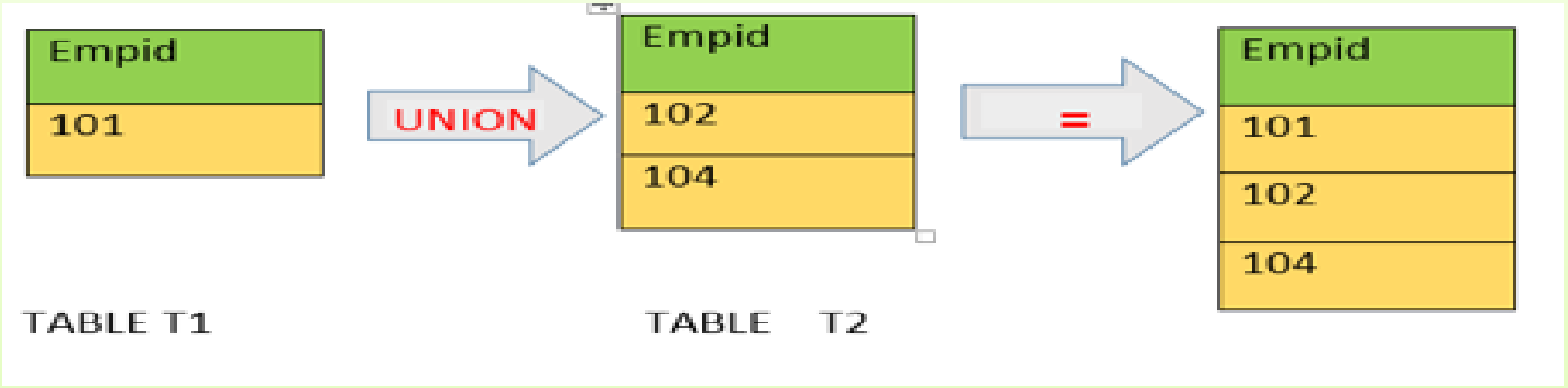
- **The MINUS compares the results of two queries and returns distinct rows from the result set of the first query that does not appear in the result set of the second query.**



- **Select Empid from T1**
- **MINUS**
- **Select Empid from T2;**

SQL – UNION

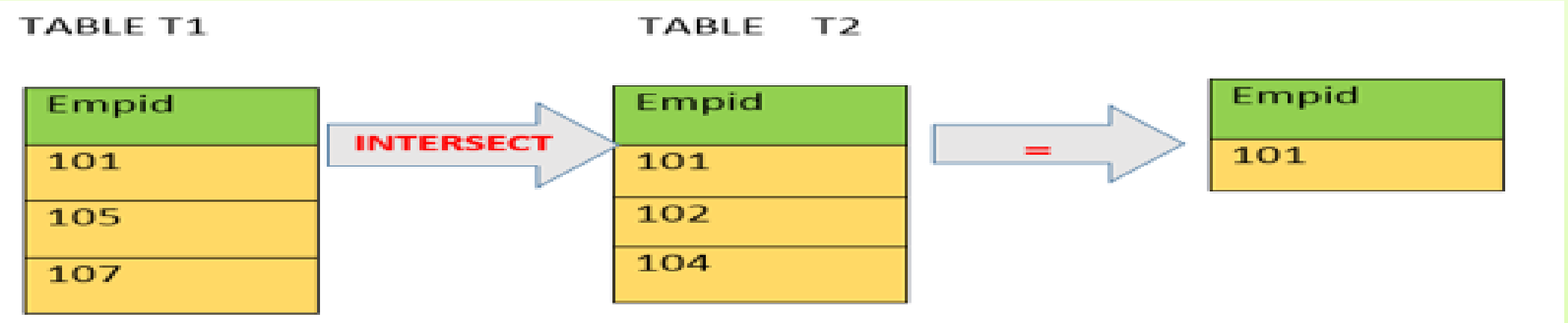
- **The UNION operator allows you to combine two or more result sets of queries into a single result set.**



- **Select Empid from T1**
- **UNION**
- **Select Empid from T2;**

SQL – INTERSECT

- **The INTERSECT operator compares the result sets of two queries and returns the distinct rows that are output by both queries.**
- **To use the INTERSECT operator for two queries, you follow these rules:**
  - **The order and the number of columns in the select list of the queries must be the same.**
  - **The data types of the corresponding columns must be compatible.**



- **Select Empid from T1**
- **INTERSECT**
- **Select Empid from T2;**