

Date.: 19-6-18
MON TUE WED THU FRI SAT

Chapter - 6

- DATA : Data is a raw fact or material that doesn't provide any meanings to the user.

For Ex: Class, Students, Teachers, benches blackboard, etc.

It Conveys inappropriate message to the user. After processing of the data it is converted into information.

- INFORMATION : The process data that provides the meaningful reason or message to the user is called as an information.

For Ex: The students of B.Tech Comp. engineering studying in a class room.

The Datas that are used to display and to perform the processing are stored in the form of tables. The tables is divided into the two dimensions that is either row and column. The rows & columns are used to store the multiple amount of data for which the datas are simplified.

- INCOMPLETE DATA : The Data that doesn't provides complete meaning or facts to the user or improve is called as incomplete data.

Ques : Which tool is used to store the large amount of data?

1. Apache Hadoop
2. Microsoft HDInsight
3. NoSQL
4. Hive
5. Sqoop
6. PolyBase
7. Bigdata in Excel
8. Presto

Oracle : Larry Ellison
Bob Miner
Ed Oates

C : Dennis Ritchie
Ken Thompson

- SQL : Structured Query Language is the Standard Command Set used to communicate with the relational database management system.
All tasks related to database management such as creating tables, modifying the data in the database

Date :
MON TUE WED THU FRI SAT

deleting them, querying the database for information giving access to the users and etc. can be done using SQL

- characteristic of SQL
It is flexible language.
It uses a Syntax that gives the user the ability to structure SQL statements in the way which is best suited to that.
Each SQL request or query is checked by relational database management system before execution for proper syntax and to optimize the request.

• Advantages of SQL
SQL is the higher level language that provides a greater degree of abstraction.

SQL enables the end-users and systems personnel to deal with a number of database management system that is available.

Applications written in SQL can be easily ported across the system.
It specifies what is required and how it should be done.

It is very simple as well as easy to learn and it can handle the complex situation.

All the SQL Operations can be performed at a Set level.

- **SQL Datatypes**

- Character
- Varchar
- time-stamp
- Integer
- Small Integer
- float
- Date (date)
- time
- Real and double decision

- **Character**: This datatype represent the fixed length of a string with exactly "n" character where $n \geq 0$ should be any integer value.

- Syntax name character(10)

$$n = 10$$

- **Varchar**: This datatype represents a varying length of string whose maximum length is "n" character.

- Syntax name Varchar(10)

- **Integer**: Integer represent assigned integer value decimal or binary

Syntax enroll_no Integer(15)

Date:
MON TUE WED THU FRI SAT

Date:
MON TUE WED THU FRI SAT

Float: A floating point number with precision of atleast "n" digits
syntax valuefloat(5).

Real and double decision: floating point value and double decision floating point number with machine dependent precision.

calendar

Date: A character date, containing the four digit number year, month & day.

Ex → date of birthdate

Time: The time of a day in hours, minutes & seconds

Ex → arrival-time time

time-stamp: The combination of date and time is represented by time-stamp.

Ex → date '2005-04-25'

time '09:10:25'

time-stamp '2005-04-25

09:10:25'

The SQL Statement is divided into the four categories

Data definition language

Data Manipulation language

Data Control language

Data Query language

Date: _____
MON TUE WED THU FRI SAT

Data Definition Language : It is used to create alter and delete database and objects. The Commands that are used are listed below

1. Create - to Create a Table
2. Alter - to perform changes
3. Drop - to delete the table.

Data Manipulation language : It allows the user to insert, modify and delete the data in the database. SQL provides three data manipulation statements.

1. Insert - to insert the row into the table.
2. Update - to perform change into the table.
3. Delete - to delete any row from the table.

Data Query language : The SQL statement enables the user to query one or more tables to get the information they needed. For this purpose SQL has only one data query statement that is Select

Data Control language : It consists of commands that controls the users to access the database objects. Various data control commands are used to update as well as to modify the database.

Date: _____
MON TUE WED THU FRI SAT

For ex: grant, revoke, abort, etc.

Datatypes : The datatypes are used to store format specific value.

Small integer : A small integer is a machine independent subset of the integer datatype that are used to store two digits only.

SQL Command

SYNTAX Create table <tablename>
(<columnName1><datatype><size>,
<columnName2><datatype><size>);
This command is used to create a new table the column names should be unique while creating a table.

For ex: Create table Student;
(stud_id Varchar(15), Stud_name
Varchar(25), stud_addr Varchar(50));

Inserting the values into the table

SYNTAX Insert into <tablename> values
(<expression1>, <expression2>, <exp-n>);

The insert statement is used to enter the rows into the table. The rows which are entered into the table should be uniquely defined by making use of any

identity.

For ex: insert into student values ('01', 'xyz', '1217');

The rows are always mapped according to the sequence of the column name.

Viewing of the data

Once the rows are inserted into the table and for viewing the rows the Select command is used to find the rows from the table.

Select * from <tablename>

- i) → Select stud_id, stud_name from student
- ii) → Select * from student where stud_id = '2';

For selecting column (i).

For selecting rows (ii),

NOTE To View the data in the database we have three Categories

- i) For all rows & all columns.
- ii) For Selected rows & all columns.
- iii) For all rows & Selected columns,

Date: _____
MON TUE WED THU FRI SAT

Date: _____
MON TUE WED THU FRI SAT

Eliminating duplicate rows by Select statement.

The distinct clause allows to remove the duplicate value from the result set. It scans through the value of columns specified and display only the unique values among them.

SELECT DISTINCT <columnname1><columnname2>
FROM <Tablename>;

The distinct clause will scan the entire rows & eliminate the rows that have exactly the same contents in each column.

SELECT DISTINCT * FROM <Tablename>;

It will display the unique rows of the table.

For ex: Select distinct * from student

How to Sort the data in the table

The rows that are retrieved from the table will be sorted either in ascending order or in descending order.

SYNTAX

Select * from <Tablename> Order by
<ColumnName1>, <ColumnName2>
<[Sort Order]>;

The Order by clause sorts the result based on the column specified. The Order by clause is used in the Select statements.

Show the details of student according to the students enroll no.

Ans → Select * from student order by
eno [ASC]

Different → Select * from student order by
eno [DESC]

Creating a Table from another table.

To Create the Sub-table from the source table we are making use of As Select Statement.

SYNTAX.

Create table <Tablename>(<ColumnName1>

DATE	MON	TUE	WED	THU	FRI	SAT

DATE	MON	TUE	WED	THU	FRI	SAT

<ColumnName2>)

As Select <ColumnName1>, <ColumnName2> FROM <Tablename>

Ex: Create table Student_mstr from a table student

Create table Student_mstr (student_eno;
student_name).

As select student_eno, student_name
From student_mstr;

* Inserting a data into a table from another table

Insert into <Tablename> SELECT
<ColumnName1>, <ColumnName2>. FROM
<Tablename>

Delete Operation

To delete a specific row from the table.

Delete from <Tablename> where Condition

Ex: Delete A student for where
enrollno = 7

Date: _____
MON TUE WED THU FRI SAT

For the removal of all the rows
Delete from <tablename>;

To delete the table from the database.
drop <tablename>;

Updating the contents of the table

Update <tablename> set <ColumnName1> =
<Expression1>, <ColumnName2> =
<Expression2>;

This statement is used for updating
all the rows in the table.

The update statement updates
the column in the existing
rows of table with the new
headings. The set clause indicates
which column data should
be modified and the new
values are inserted in that
column.

Ex : Update the address details
(Modify, Change) of the
Student by changing its city
name to Bombay.

Update Student set address =
'Bombay';

Date: _____
MON TUE WED THU FRI SAT

Updating records conditional where
claws is used.

Update <tablename> Set <ColumnName1> =
<Expression1>, <ColumnName2> =
<Expression2> where <condition>.

Ex : Update the student details.
by changing Bombay to Surat.

update <student> set address = Surat;

Modifying the structures of table.

The structure of the table can be
modified by using 'alter' command.
'ALTER' Table allows changing the structure
of an existing table. With 'alter' table,
it is possible to add or to delete the
columns. To create or to destroy the
indexes, to change the datatype of
existing columns or to rename the
columns or the table itself.

SYNTAX FOR ADDING THE NEW COLUMNS.

ALTER Table <Table name> ADD (<New Colu-
mn Name> <Datatype> (<size>),
<New Column Name> <Datatype> (<size>),
.....);

Eg: Modify or Enter a new field (column) called city in the table Student mstr.

ALTER Table Student mstr ADD (city varchar(100)).

Dropping a column from the Table

SYNTAX

ALTER Table <Table name> DROP Column <column Name>;

Eg: Drop the column city from student mstr table

ALTER Table Student mstr DROP column city;

Modifying the existing columns.

SYNTAX

ALTER Table <Table name> MODIFY (<column name> <New Datatype> (New size));

Eg:

ALTER Student mstr table to allow the name field to hold maximum of 30 char.

Date:
MON TUE WED THU FRI SAT

ALTER
MODIFY Table student mstr MODIFY
student name varchar(30);

Rules / Restriction for using ALTER Table

The foll. task cannot be perform by using ALTER Table claws.

1. To change the name of a table.
2. To change the name of a column.
3. To decrease the size of the column if the table data exist.

Creating / Renaming

Rename the name of the table.

SYNTAX

RENAME <Table name> to <New-Table name>;

Eg: Change the name of Student mstr table to Student table

Rename student mstr to student

Truncating the table

Truncate Table is used to empty a table logically. This is equivalent to a delete statement that deletes all the rows.

By using the truncate keyword, table the number of deleted rows are not returned as it uses make use of primary key constraint.

Truncate table <Tablename>

Synonyms

A Synonyms is an alternative name for the objects such as tables, views, sequences, stored procedure and other database object.

SYNTAX:

Create [or Replace] [Public] SYNONYM
[SCHEMA] SYNONYM_NAME FOR [SCHEMA]
object_name [@DBLINK]

Examine to objects created by a user

Select * from TAB;

To display the table structure

Describe <Tablename>
DESC

Show the table structure of student_mstr

Describe student_mstr

Data Constraint

Data Constraints are the set of rules that must be applied to the data which is gathered, stored or analyzed in the database or table to ensure its integrity. The rules applying to the data will be different for the different applications. There are two types of data constraints that are applied to the data being inserted into the oracle table. One type of constraint is I/O constraint, and the other type of constraint is business rule constraint.

SYNTAX:

PRIMARY KEY.

<columnName> <datatype> <size>
primary key

A primary key is one or more columns used to uniquely identify each row in the table. None of the fields that are the part of primary key can contain a null value.

Date: _____
MON TUE WED THU FRI SAT

A table can have only one primary key.

A single column primary key is called as a simple key.

A multicolumn primary key is called as a composite primary key. The only form of a primary key in a table is to uniquely identify a row.

Features of Primary key

- The main purpose of the primary key is to record uniqueness.
- It will not allow duplicate values.
- It will not allow the null values.
- It is not compulsory but it is recommended.
- Only one primary key is allowed per table.

Primary Key Constraint defined at table level.

SYNTAX

PRIMARY KEY (<columnName>
<columnName>)

Eg: Insert the primary key for the whole table Student_mstr with the fields

Date: _____
MON TUE WED THU FRI SAT

student_eno, student_name and
student_add

Create table student_mstr (student_eno varchar(20), student_name varchar(25), student_add varchar(50), primary key (student_eno, student_name, student_add))

Foreign Key Constraint

The foreign key represents relationship between the two tables. A foreign key is a column or the group of columns whose values are derived from the primary key or unique key of some other table. The table in which foreign key is defined is called as foreign table or detail table. The table that defines the primary or unique key & it is referenced by the foreign key is called primary table or master table.

SYNTAX : <columnName> <Datatype>
(<size>) REFERENCES <Tablename>
[(<columnName>)] [ON DELETE CASCADE]

Eg: Create a Table Student_mstr with its primary key as student_eno referencing the foreign key course_no in Student table.

Date: 13-7-18
MON TUE WED THU FRI SAT

Create table Student

Course_no Varchar(20) Reference
Student [student_en_no]

Foreign Key Constraint at table level.

Foreign Key (<columnName>, <columnName>,
REFERENCES <TableName>[<columnName>]
<<columnName>>])

Eg: Create table student details with
Course no as foreign key reference
in the column of course_no in the
table student_mstr

Create table student_details (student_id
Varchar(10), course_no Varchar(10),
Foreign Key (course_no) REFERENCES
Student_mstr (course_no));

Foreign key constraint define with
on delete Cascade.

Create a table student details with
its foreign key as student serial no
with On Delete Cascade option

Create table Student

Date:
MON TUE WED THU FRI SAT

Unique Key

The unique Column Constraint permits
multiple entries of null into the
column. The null values are joint together
or collected at the top of the
Column in the Order in which they
are enter into the Table. This is
the primary difference between the
primary key & the unique consta-
aint applied to the Table columns.

<columnName> <Datatype> (<size>)
UNIQUE

Unique key Constraint Defined at
table level

Create table tablename (<columnName1>
<Datatype> (<size>) UNIQUE
<columnName1>, <columnName2>));

Create table Student_mstr such that
the Contains of the Column student
id are unique across the entire
column.

Create table S:

Null Value Concepts

- A Null Value is used when the actual value is unknown or when the value is not meaningful.
- A Null Value is not equivalent to a value of zero if the datatype is number & it is not equivalent to space if the datatype is char.
- A Null Value will evaluate to null in any expression.
- A Null value can be inserted into columns of any datatype.

Difference between an empty string and a null value.

Select * from Student_mstr where Name = " ";

→ While this statement is executed it is expected to return the row that is inserted about but this statement will not retrieve any record.

Select * from Student_mstr where Name = Null ;

→ When this statement is executed the rows are retrieved from the table.

Date:
MON TUE WED THU FRI SAT

Date:
MON TUE WED THU FRI SAT

NotNull Constraint Defined at the Table level.

Oracle has the Notnull Constraint defined at its Column level Constraint. The notnull Constraint ensures that a table column cannot be left empty when a column is defined as not null then that column becomes the mandatory column.

SYNTAX

<ColumnName> <Datatype> (<size>)
NOTNULL

CHECK CONSTRAINT

Business rule Validation can be applied to a table column by using the check Constraint. Check Constraint must be specified as the logical expression that evaluates either True or False.

SYNTAX

At Column Level

<ColumnName> <Datatype> (<size>)
CHECK (<logical expression>)

At table level

CHECK (<logical expression>)

Date: MON TUE WED THU FRI SAT

Create a student mstr with the full. Check Constraint

- 1) Data values inserted into the column course no must start with the capital letter C.
- 2) Data value being inserted into the column Ename, M_name, L_name should be in the upper case only.

SYNTAX :

CHECK (course_no LIKE 'C%')
CHECK (Ename = Upperc(Ename))

Computation done on TABLE DATA

The Computations done on table data is done by using arithmetic operation & logical operators

Arithmetic Operators

Data manipulation such as insert, update & delete can also be performed by using arithmetic operators such as

- + Addition
- Subtraction
- * Multiplication
- / Division

- ** Exponent
- () Enclosed bracket

Logical Operators

AND : The AND Operator allows for creating the SQL statements based on two or more conditions. It can be used in any SQL statements such as Select, Insert, Update or Delete.

Display all those transaction performed today for the amount ranging betn 500 - 5000 ; Amount, Period, Interest Rate
Select * from Trans_mstr where amt >= 500 and amt <= 5000
and To_char (DT, DD/MM/YYYY)

CHAPTER -

Date: 18/7/18
MON TUE WED THU FRI SAT

Computation Done On Table Data

The Computations done on table data is done by using arithmetic operation & logical operators

Arithmetic Operators

Data manipulation such as insert, update & delete can also be performed by using arithmetic operators such as

- + Addition
- Subtraction
- * Multiplication
- / Division

** Exponent

() Enclosed bracket

Logical Operators

AND : The AND Operator allows for creating the SQL statements based on two or more conditions. It can be used in any SQL statements such as Select, Insert, Update or Delete.

Display all those transaction performed today for the amount ranging betn 500 - 5000 ; Amount, Period, Interest Rate
Select * from Trans_mstr where amt >= 500 and amt <= 5000
and To_char (DT, DD/MM/YYYY)

OR: The OR condition requires that any of the condition must be satisfied for the record to be included in the result set.

Display the customer that belongs to, I.T or are self employed
Cust_mstr, cust_no, Fname, L_name, occ

Select * from customer_mstr where occ = 'I.T' OR occ = 'self employed'.

Combining AND and OR Operators:
The Oracle engine will process all the rows in a table & display the result only when all of the conditions specified using the AND operator are satisfied & when any of the conditions specified using the OR operator is satisfied.

NOT Operator

The Oracle engine will process all the rows in a table & display only those records that do not satisfy the conditions specified.

List the accounts details of those accounts which are neither singly nor joint accounts
Account_mstr

Days	SUN	MON	TUE	WED	THU	FRI	SAT

Acc_no, Type, OPP_Mode, Status

Select * from Account_mstr where NOT (OPP_Mode = 'Singly' OR OPP_Mode = 'Joint');

Range Searching

To select the data that is specified within the range of values the Between Operators is used.

List the transactions performed in the month of Jan. to March

Trans_mstr

Amount, Period, Interest_Rate

Select * from Trans_mstr where To_char(DT, 'MM') Between 01 AND 03

Select * from Trans_mstr where To_char(DT, 'MM') >= 01 AND To_char(DT, 'MM') <= 03

NOT Bet'n Operator

The Oracle engine will display the result according to the conditions not specified.

Pattern Matching

The Like predicate allows the comparison of one string value with another string value which is not

Date:
 MON TUE WED THU FRI SAT

identical or similar. This can be done by using different characters
 1. % allows to match any string of any length (including 0 length)
 2. _ allows to match on a single character

List the customer's name that begins with the letters 'Ch'.

Cust_mstr
F-name, L-name, DOB

Select * from cust_mstr where
(F-name LIKE 'Ch%');

List the customers names that have the second character as 'c' or 's'

Select * from cust_mstr where
(F-name LIKE 'a%' OR 's%');

List the customers names that begin with the letters 'TV' & it is a fourth letter word.

Select * from cust_mstr where
(F-name LIKE 'TV%');

IN, NOT And NOT IN OPERATOR

The arithmetic operator equal to $= 1$
Compares a single value to the another single values. In case a value needs to be compared to the list of values then `IN predicate` is used. The `IN predicate` helps to reduce the need of multiple `OR` condition.

List the customer details of the customer name Hasel, Mamta, Namita, ARUNA.

Cust_mstr (F-name, L-name, Occup)
Select * from cust_mstr where F-name
IN ('Hasel', 'Mamta', 'Namita', 'ARUNA')

NOT-IN predicate

The `not-in predicate` is the opposite of `in predicate`. This will select all the rows where the values do not match with the values in the list.

List the customer details of the customer than Hasel, Mamta, Namita, Aruna

Select * from cust_mstr where
F-name NOT IN (

ORACLE Table - DUAL

Select * from DUAL
columnname NULL Datatype

SYSTEM . DATE

System date is a column that contains the current date & time with no arguments.

Select SYSDATE from DUAL
OUTPUT : 20-July-18

ORACLE FUNCTIONS

Aggregate Functions

1. String Functions
2. Date Functions.

Aggregate functions or the numeric functions are used for number datatypes.

' Returns the average value of 'n' By ignoring the null values in a column.

Avg :- 'n'

Avg ([<DISTINCT> | <ALL>] <n>)

Select Avg (Price) from Cust_mstr;

MINIMUM

Date:
MON TUE WED THU FRI SAT

Date:
MON TUE WED THU FRI SAT

Select MIN (Price) from cust_mstr;

It will return the minimum values.

MAXIMUM

Select MAX (Price) from cust_mstr;
It will return the maximum values.
for all columns

COUNT

Select Count (Acc_no) from Cust_mstr;

SUM

Select SUM (Price) from cust_mstr;

Numeric Functions

1) AVG(n) :- Absolute

Select AVG(-15) from DUAL

OUTPUT : → 15

2) POWER(m,n) :- Power

Select POWER(3,2) from DUAL

OUTPUT : → 9

3) ROUND (n, [p,m])

Select ROUND(15.19,1) from DUAL

OUTPUT : → 15.2

4) SQRT(n) :- Square root

Select SQRT(25) from DUAL

OUTPUT : → 5

5) GREATEST(expr1,expr2,expr3,...)

Select GREATEST (1,5,17) from DUAL

OUTPUT : → 17

- 6) LEAST($exp_1, exp_2, exp_3, \dots$)
 Select LEAST(1, 5, 17) from DUAL.
 OUTPUT: 1
- 7) MOD(m, n) :- modulus (%)
 Select MOD(15, 7) from DUAL
 OUTPUT: 1
- 8) TRUNC([number, [decimal-places]]) :-
 Truncate
 Select TRUNC(125.815, 1) from DUAL
 OUTPUT: 125.8
- 9) FLOOR(n)
 Select FLOOR(24.8) from DUAL
 OUTPUT: 24
 It returns the largest integer value that is equal to or less than a no.
- 10) CEIL(n)
 It returns the smallest integer value that is greater than or equal to a no.
 Select CEIL(24.8) from DUAL
 OUTPUT: 25

⇒

- i) LOWER .
 It returns the character with all the letter in lowercase.

- Date : MON TUE WED THU FRI SAT
- String Function
- a) lower(char)
 Select lower('Ivan Bayrass') From dual;
- b) INITCAP First letter should be capital. It return a string with the first letter of each word in uppercase.
 Select INITCAP ('ivan bayrass') From dual;
- c) upper(char)
 It returns characters with all the letter force to uppercase.
 Select UPPER ('Ms. Carol') From dual;
 Output: MS. CAROL
- d) SUBSTR
 It returns the position or the portion of the character beginning at the character 'm' & going upto the character 'n'.
 Select SUBSTR(<string>, <start position>, <length>)
 Select SUBSTR('SECURE', 3, 4) from dual;
- e) ASCII
 It returns the number code that represent the specified character.
 If more than one character is entered the funb will return the value for the first character &

Date: _____
MON TUE WED THU FRI SAT

ignore all the characters after the first.

ASCII (single character)

Select ASCII ('A'), ASCII('a') from dual;

Output: A → 65
a → 97

f) COMPOSE

It returns the unicode string. It can be either a character or Varchar, etc.

COMPOSE (single)

Select COMPOSE ('a') || UNISTR ('\u0301')
from dual;

g) INSTR

It returns the location of a substring in a String.

INSTR ('String', 'start position'),
<nth appearance

Select INSTR ('SCT on the net', 't') // 8

INSTR ('SCT on the net', 't', 1, 2)
from dual; // 14

h) i) TRANSLATE

TRANSLATE ('String1', 'String2 Replace',
'Replacement');

It replaces a sequence of characters in a string with another set of characters.

Date: _____
MON TUE WED THU FRI SAT

Select TRANSLATE ('1sct523', '123', '7aq')
from dual;

Output: 7sct5aq

i) LENGTH
Length (word)

j) LTRIM

It removes the character from the left of characters with initial characters removed upto the first character not in the Set.

LTRIM (char, [set])

Select LTRIM ('xyz', 'x') from dual;

Output: yz

k) RTRIM

It returns the character after removes from the right of characters with initial characters removed upto the first character not in the Set.

RTRIM (char [set])

Select RTRIM ('xyz', 'z') from dual;

Output: xy

l) TRIM

It removes all the specified characters either from the beginning or ending.

SYNTAX

`TRIM ([leading] [trailing] [both] <trim character>
FROM1] <string1>)`

Eg: `Select TRIM (' Hansel ') from Dual;`

O/P :- Hansel

`Select TRIM (Leading X from 'xxx Hansel xxx')
from Dual;`

O/P :- Hanselxxx

`Select TRIM (Both X from 'xxx Hansel xxx')
from dual;`

O/P :- Hansel

`Select TRIM (Both '1' from '123 Hansel 12111')
from dual`

m) Left Padding [LPAD]

→ It returns character | left Padded to the length 'n' with sequence of character specified with character

SYNTAX

`Select LPAD (char1, n [char2])`

Eg:

`Select LPAD ('Page1', 10 '*') from Dual;`

O/P :-

***** Page 1

Date:
MON TUE WED THU FRI SAT

Date:
MON TUE WED THU FRI SAT

n) RPAD (Right Padding)

`Select RPAD ('Ivan', 10, 'x') from Dual;`
Output: Ivanxxxxxx

VSIZE

It returns the no. of bytes for the internal representation of the expression.

`Select VSIZE ('TELNET') from dual;`
Output: 6

o) Conversion Function

It converts a character or the character value expressing a number to a number datatype

1) SYNTAX To_number (char)

Eg: `Update Student_mstr Set {DOB = + To_number ('Substs'(''$100'), 2, 3)};`
Output: 10 - \$100 Jan - 2011

2) To_Char (n, fmt)

`Select To_Char (17145, '$099,999') from Dual.`

Output: \$017145

p) Date Conversion Function

1) To_Date

It converts the character field to date field.

SYNTAX `To_Date (char, fmt)`

Date: MON TUE WED THU FRI SAT

Insert into stud_details (std_no, Enam, M_name, L_name, DOB) values
 Values ('S1', 'Ivan', 'Nelson', 'Bayross');
 To Date ('25-JUN-1952' 10:55 AM',
 'DD-MM-YYYY HH:MI AM'));

2) ADD_MONTHS

It returns the date after adding the no. of months Specified in the funn.

SYNTAX: Add_months (System_date, 4)
 from dual;

Output: 27-Nov-2018

3) LAST_DAY

It returns the last date of the month Specified within the funn.

SYNTAX: LAST_DAY (DAY)

Select SYSDATE, LAST_DAY (SYSDATE)
 from Dual;

Output: 31st - July - 2018

Months betⁿ, Next day, Round

Date: MON TUE WED THU FRI SAT

Grouping of Data in SQL

Group By - The Group By clause creates a data set containing several sets of record grouped together on the basis of some specified condition

SYNTAX

Select <ColumnName1>, <ColumnName2>, <ColumnNameN> Aggregate function (<expression>) from tablename where <Condition>
 Group by <Columnname1><ColumnNames> <Columnname N>

Find out how many employees are there in each branch.

emp_mstr (branch_no, emp_no).

B1	1
B2	2
B3	3
B4	4
B5	5

Select <branch_no>, Count (Kemp_no)
 from emp_mstr Group by <branch_no>;

Find Out the total no. of current & Savings bank account Verified by each employee.

ACCT_MSTR (Verify_Acc, Acc_no, emp_no)

Date:
MON TUE WED THU FRI SAT

Select Verify_Act_Emp, Count(Acc_no)
from Acct_Mstr Group by
Verify_Act_Emp

Find Out Total no. of Accts Separated
on the basis of acc. type per branch.
Acct_mstr(Br_no, Acc_no, type)

Select type, Count(Acc_no) from
Acct_mstr Group by type, Br_no

Having clause

The having clause is used with
group by clause. Having clause imposes
condition on the group by clause which
further filters the groups created by
the group by clause.

Find out the customers having more than
one acbt in a bank.

Acct_Fd_cust_dtls(cust_no, Acc_Fd_no)

Select cust_no, Count(Acc_Fd_no) from
Acct_Fd_cust_dtls where Group by
cust_no having Count(Acc_Fd_no) > 1.

Find out the no. of accounts opened at
a branch after 3rd-Jan-2003 only if
the no. of acct open after 3-Jan-2003
exceed 1.

Date:
MON TUE WED THU FRI SAT

Acct_mstr(Br_no, Acc_no) OPNDT

Select Br_no

Select Br_no, Count(Acc_no) from
Acct_mstr where (OPNDT::to_char
(OPNDT, 'DD-MM-YYYY')) > '03-Jan-2003'
group by branch_no having count(Acc_no)
> 1;

List out the custno which are associated
with only one acct. in the bank
Acct_Fd_cust_dtls(cust_no, Acc_Fd_no)

FD-MSTR (FD_Ser_No, FD_Name,

Select FD_Ser_No, FD_No, SUM(Amt),
SUM(DueAmt) FROM FD-MSTR Group by
Rollup (FD_Ser_No, FD_No)

FD_Ser_No	FD_No	Amt	Due_Amt
F1	F1	10,000	11,000
	F2	20,000	22,000
	F3	10,000	11,000
FS1	3FD	40,000	44,000

Group by using Cube operator

The Cube operator can be applied to all the aggregate functions like avg(), avg(), count(), sum(), max(), min() with in a group by statement. The Cube operator produces sub totals for all possible combination of the grouping specified in the group by clause with a grand total as opposite of roll up operator which produces only a fraction of possible sub total combinations.

Find out the balance of the account holders on per account & per branch basis along with the grand total.

Date: _____
MON TUE WED THU FRI SAT

Acc_mstr (Branch_no, Acc_no, Sum_of_Curbal)

Select Branch_no, Acc_no, Sum(Sum_of_Curbal) from Acc-mstr Group by
Cube (Branch_no, Acc_no);

Branch_no	Acc_no	Sum(Sum_of_Curbal)
B1	CAT	23000
B1	SBI	22000
B1	SB11	500
B1	SB11	500

Subqueries

A Subquery is a form of SQL statement that appears inside another SQL statement. It is also being termed as nested query. The statement containing a subquery is called as the parent statement. The parent statement uses the rows written by the subqueries.

- purpose of the subqueries
- 1. To insert the records in the target table.
- 2. To create the tables and to insert the record in the table created.
- 3. To update the records in the table created or the target table.
- 4. To create the views.
- 5. To provide the values for

Conditions in where clause, having clause, impredicate and it is used with select, update and delete statements

6. It is used in reference with parent table.

Retrive the address of a customer named 'Ivan_Bayross'

cust_mstr, Add_details
(cust_no, F_name, L_name), (Add1, Add2,
city, state, pincode)

Select Add1, Add2,

Select custno, Add1, Add2, state, city,
Pincode from Address_details where
cust_no IN (Select cust_no from
cust_mstr where F_name = 'Ivan' and
L_name = 'Bayross'));