ORACLE components and commands

Oracle Components

I. Oracle DBMS

A. Oracle Display System (ODS) - activities in the system displayed on the system console.

B. Oracle Data Loader (ODL) - loads raw data into tables.

C. After Image Journaling (AIJ) - program for journaling each change to data in case of need for recovery.

D. System Global Information (SGI) - shows shared memory set up.

E. IOR - program for start and stopping ORACLE system for the DBA. This contains the setups and saves.

F. Export/Import - for use to restore of backup archival storage.

II. Pro\*SQL - host language interface - group of subroutines which translates SQL statements into machine code.

III. SQL\*Plus - include SQL and SQL\*Plus commands.

IV. Oracle Report Writer:

A. Report text formatter - similar to a word processor.

B. Report generator - procedural language that creates a report from the data base.

V. SQL\*FORMS - interactive facility to create and edit forms (screens).

A. IAG - Interactive Applications Generator.

B. IAC - Interactive Applications Converter.

C. IAD - Interactive Applications Designer.

D. IAP - Intaeractive Applications Processor.

E. Crt Utility - for different types of displays.

VI. Audit- Trail - enables DBA to monitor activities in system for security or tuning purposes.

VII. SORT/MERGE - to improve efficiency.

The commands are listed in the following order:  SQL\*Plus commands,

PL/SQL commands, and then SQL commands.

SQL\*PLUS COMMANDS

     @                        @@                     /

     ACCEPT                   APPEND                 BREAK

     BTITLE                   CHANGE                 CLEAR

     COLUMN                   COMPUTE                CONNECT

    COPY                     DEFINE                 DEL

    DESCRIBE                 DISCONNECT             EDIT

    EXECUTE                  EXIT                   GET

    HELP                     HOST                   INPUT

    LIST                     PAUSE                  PRINT

    PROMPT                   REMARK                 RUN

    RUNFORM                  SAVE                   SET

    SHOW                     SPOOL                  SQLPLUS

    START                    TIMING                 TTITLE

    UNDEFINE                 VARIABLE               WHENEVER OSERROR

    WHENEVER SQLERROR

PL/SQL COMMANDS (Statements)

    CLOSE Statement          EXIT Statement         FETCH Statement

    GOTO Statement           IF Statement           LOOP Statement

    NULL Statement           OPEN Statement         RAISE Statement

    RETURN Statement

SQL COMMANDS

The SQL commands are divided into these categories:

    \* Data Definition Language commands

    \* Data Manipulation Language commands

    \* Transaction Control commands

    \* Session Control commands

    \* System Control commands

SQL Data Definition Language commands include the following:

    ALTER CLUSTER            ALTER SEQUENCE         COMMENT

    ALTER DATABASE           ALTER SNAPSHOT         CREATE CLUSTER

    ALTER FUNCTION           ALTER SNAPSHOT LOG     CREATE CONTROLFILE

    ALTER INDEX              ALTER TABLE            CREATE DATABASE

    ALTER PACKAGE            ALTER TABLESPACE       CREATE DATABASE LINK

    ALTER PROCEDURE          ALTER TRIGGER          CREATE FUNCTION

    ALTER PROFILE            ALTER USER             CREATE INDEX

    ALTER RESOURCE COST      ALTER VIEW             CREATE PACKAGE

    ALTER ROLE               ANALYZE                CREATE PACKAGE BODY

    ALTER ROLLBACK SEGMENT   AUDIT                  CREATE PROCEDURE

    CREATE PROFILE           DROP ROLE

    CREATE ROLE              DROP ROLLBACK SEGMENT

    CREATE ROLLBACK SEGMENT  DROP SEQUENCE

    CREATE SCHEMA            DROP SNAPSHOT

    CREATE SEQUENCE          DROP SNAPSHOT LOG

    CREATE SNAPSHOT          DROP SYNONYM

    CREATE SNAPSHOT LOG      DROP TABLE

    CREATE SYNONYM           DROP TABLESPACE

    CREATE TABLE             DROP TRIGGER

    CREATE TABLESPACE        DROP USER

    CREATE TRIGGER           DROP VIEW

    CREATE USER              GRANT

    CREATE VIEW              NOAUDIT

    DROP CLUSTER             RENAME

    DROP DATABASE LINK       REVOKE

    DROP FUNCTION            TRUNCATE

    DROP INDEX               UPDATE

    DROP PROCEDURE

    DROP PROFILE

SQL Data Manipulation Language commands include the following:

    DELETE

    EXPLAIN PLAN

    INSERT

    LOCK TABLE

    SELECT

SQL Transaction Control commands include the following:

    COMMIT

    ROLLBACK

    SAVEPOINT

    SET TRANSACTION

SQL Session Control commands include the following:

    ALTER SESSION

    SET ROLE

SQL System Control command (only one command):

    ALTER SYSTEM

Insert Command

INSERT INTO t/v\_name (column, column,...)    --(column list) optional

    VALUES(value, value,...);

 INSERT INTO t/v\_name (column, column,...) SELECT...;

 INSERT adds rows to the table or view specified.  Names in the column

 list may be in any order.  Unlisted columns are set to NULL; you must

 list any column created NOT NULL, and supply a value for it.

 VALUES inserts each item in its list of values into the corresponding

 column in the list of columns. The datatypes of each pair of items in

 the two lists must be compatible or convertible. CHAR and DATE values

 must be enclosed in 'single quotes'.

 Use a subquery to INSERT rows from another table.  SELECT must return

 a value for each column listed.

Update Command

UPDATE table/view\_name SET

     column\_name = sql\_expression                       --SQL option

     column\_name = (one\_value\_SELECT\_statement)         --SQL option

     (column\_name, column\_name,...) = SELECT\_statement  --SQL option

   WHERE\_clause;

 UPDATE replaces the values in rows of one or more specified columns

 with the values from a SQL expression or a SELECT statement. SELECT

 must return at least one row, and every column to the left of the =

 sign.  SELECT may not contain an INTO clause.  Use the WHERE clause

 to specify conditions rows must meet in order to be updated; if you

 omit the WHERE clause, all rows are updated.

DELETE FROM table\_or\_view  WHERE clause;

 DELETE removes one or more rows from the table or view you specify.

 The WHERE clause specifies the rows to remove.  To remove all rows,

 omit the WHERE clause.

SELECT command

PURPOSE:

    To retrieve data from one or more tables, views, or snapshots.

SYNTAX:

SELECT [DISTINCT | ALL] { \*

                        | { [schema.]{table | view | snapshot}.\*

                          | expr [c\_alias] }

                       [, { [schema.]{table | view | snapshot}.\*

                          | expr [c\_alias] } ] ... }

    FROM [schema.]{table | view | snapshot}[@dblink] [t\_alias]

      [, [schema.]{table | view | snapshot}[@dblink] [t\_alias] ] ...

    [WHERE condition ]

    [ [START WITH condition] CONNECT BY condition]

    [GROUP BY expr [, expr] ... [HAVING condition] ]

    [{UNION | UNION ALL | INTERSECT | MINUS} SELECT command ]

    [ORDER BY {expr|position} [ASC | DESC]

           [, {expr|position} [ASC | DESC]] ...]

    [FOR UPDATE [OF [[schema.]{table | view}.]column

                 [, [[schema.]{table | view}.]column] ...] [NOWAIT] ]

where:

DISTINCT

    returns only one copy of each set of duplicate rows selected.

    Duplicate rows are those with matching values for each expression in

    the select list.

ALL

    returns all rows selected, including all copies of duplicates.

    The default is ALL.

\*

    selects all columns from all tables, views, or snapshots listed in

    the FROM clause.

table.\*

view.\*

snapshot.\*

    selects all columns from the specified table, view, or snapshot.

    You can use the schema qualifier to select from a table, view, or

    snapshot in a schema other than your own.

    If you are using Trusted ORACLE, the \* does not select the ROWLABEL

    column.  To select this column, you must explicitly specify it in

    the select list.

expr

    selects an expression, usually based on columns values, from one of

    the tables, views, or snapshots in the FROM clause.  A column name

    in this list can only contain be qualified with schema if the table,

    view, or snapshot containing the column is qualified with schema in

    the FROM clause.

c\_alias

    provides a different name for the column expression and causes the

    alias to be used in the column heading.  A column alias does not

    affect the actual name of the column.  Column aliases cannot be

    referenced elsewhere in a query.

schema

    is the schema containing the selected table, view, or snapshot.  If

    you omit schema, ORACLE assumes the table, view, or snapshot is in

    your own schema.

table

view

snapshot

    is the name of a table, view, or snapshot from which data is

    selected.

dblink

    is complete or partial name for a database link to a remote database

    where the table, view, or snapshot is located.  Note that this

    database need not be an ORACLE7 database.

    If you omit dblink, ORACLE assumes that the table, view, or snapshot

    is on the local database.

t\_alias

    provides a different name for the table, view, or snapshot for the

    purpose of evaluating the query and is most often used in a

    correlated query.  Other references to the table, view, or snapshot

    throughout the query must refer to the alias.

WHERE

    restricts the rows selected to those for which the condition is

    TRUE.  If you omit this clause, ORACLE returns all rows from the

    tables, views, or snapshots in the FROM clause.

START WITH

CONNECT BY

    returns rows in a hierarchical order.

GROUP BY

    groups the selected rows based on the value of expr for each row and

    returns a single row of summary information for each group.

HAVING

    restricts the groups of rows returned to those groups for which the

    specified condition is TRUE.  If you omit this clause, ORACLE

    returns summary rows for all groups.

UNION

UNION ALL

INTERSECT

MINUS

    combines the rows returned by two SELECT statement using a set

    operation.

ORDER BY

    orders rows returned by the statement.

            expr

                   orders rows based on their value for expr.  The

                   expression is based on columns in the select list or

                   columns in the tables, views, or snapshots in the

                   FROM clause.

            position

                   orders rows based on their value for the expression

                   in this position of the select list.

            ASC

            DESC

                   specifies either ascending or descending order.  ASC

                   is the default.

FOR UPDATE

    locks the selected rows.

NOWAIT

    returns control to you if the SELECT statement attempts to lock a

    row that is locked by another user.  If you omit this clause, ORACLE

    waits until the row is available and then returns the results of the

    SELECT statement.

PREREQUISITES:

    For you to select data from a table or snapshot, the table or

    snapshot must be in your own schema or you must have SELECT

    privilege on the table or snapshot.

    For you to select rows from the base tables of a view, the owner of

    the schema containing the view must have SELECT privilege on the

    base tables.  Also, if the view is in a schema other than your own,

    you must have SELECT privilege on the view.

    The SELECT ANY TABLE system privilege also allows you to select data

    from any table or any snapshot or any view's base table.

    If you are using Trusted ORACLE in DBMS MAC mode, your DBMS label

    must dominate the creation label of each queried table, view, or

    snapshot or you must have READUP system privileges.

SEE:

    DELETE, UPDATE

CREATE TABLE command

PURPOSE:

    To create a table, the basic structure to hold user data, specifying

    this information:

    \* column definitions

    \* integrity constraints

    \* the table's tablespace

    \* storage characteristics

    \* an optional cluster

    \* data from an arbitrary query

SYNTAX:

CREATE TABLE [schema.]table

     ( { column datatype [DEFAULT expr] [column\_constraint] ...

       | table\_constraint}

    [, { column datatype [DEFAULT expr] [column\_constraint] ...

       | table\_constraint} ]...)

    [ [PCTFREE  integer] [PCTUSED  integer]

      [INITRANS integer] [MAXTRANS integer]

      [TABLESPACE tablespace]

      [STORAGE storage\_clause]

    | [CLUSTER cluster (column [, column]...)] ]

    [ ENABLE   enable\_clause

    | DISABLE disable\_clause ] ...

    [AS subquery]

where:

schema

    is the schema to contain the table.  If you omit schema, ORACLE

    creates the table in your own schema.

table

    is the name of the table to be created.

column

    specifies the name of a column of the table.  The number of columns

    in a table can range from 1 to 254.

datatype

    is the datatype of a column.

DEFAULT

    specifies a value to be assigned to the column if a subsequent

    INSERT statement omits a value for the column.  The datatype of the

    expression must match the datatype of the column.  A DEFAULT

    expression cannot contain references to other columns, the

    pseudocolumns CURRVAL, NEXTVAL, LEVEL, and ROWNUM, or date constants

    that are not fully specified.

column\_constraint

    defines an integrity constraint as part of the column definition.

table\_constraint

    defines an integrity constraint as part of the table definition.

PCTFREE

    specifies the percentage of space in each of the table's data blocks

    reserved for future updates to the table's rows.  The value of

    PCTFREE must be a positive integer from 1 to 99.  A value of 0

    allows the entire block to be filled by inserts of new rows.  The

    default value is 10.  This value reserves 10% of each block for

    updates to existing rows and allows inserts of new rows to fill a

    maximum of 90% of each block.

    PCTFREE has the same function in the commands that create and alter

    clusters, indexes, snapshots, and snapshot logs.  The combination of

    PCTFREE and PCTUSED determines whether inserted rows will go into

    existing data blocks or into new blocks.

PCTUSED

    specifies the minimum percentage of used space that ORACLE maintains

    for each data block of the table.  A block becomes a candidate for

    row insertion when its used space falls below PCTUSED.  PCTUSED is

    specified as a positive integer from 1 to 99 and defaults to 40.

    PCTUSED has the same function in the commands that create and alter

    clusters, snapshots, and snapshot logs.

    The sum of PCTFREE and PCTUSED must be less than 100.  You can use

    PCTFREE and PCTUSED together use space within a table more

    efficiently.

INITRANS

    specifies the initial number of transaction entries allocated within

    each data block allocated to the table.  This value can range from 1

    to 255 and defaults to 1.  In general, you should not change the

    INITRANS value from its default.

    Each transaction that updates a block requires a transaction entry

    in the block.  The size of a transaction entry depends on your

    operating system.

    This parameter ensures that a minimum number of concurrent

    transactions can update the block and helps avoid the overhead of

    dynamically allocating a transaction entry.

    The INITRANS parameter serves the same purpose in clusters, indexes,

    snapshots, and snapshot logs as in tables.  The minimum and default

    INITRANS value for a cluster or index is 2, rather than 1.

MAXTRANS

    specifies the maximum number of concurrent transactions that can

    update a data block allocated to the table.  This limit does not

    apply to queries.  This value can range from 1 to 255 and the

    default is a function of the data block size.  You should not change

    the MAXTRANS value from its default.

    If the number concurrent transactions updating a block exceeds the

    INITRANS value, ORACLE dynamically allocates transaction entries in

    the block until either the MAXTRANS value is exceeded or the block

    has no more free space.

    The MAXTRANS parameter serves the same purpose in clusters,

    snapshots, and snapshot logs as in tables.

TABLESPACE

    specifies the tablespace in which ORACLE creates the table.  If you

    omit this option, then ORACLE creates the table in the default

    tablespace of the owner of the schema containing the table.

STORAGE

    specifies the storage characteristics for the table.  This clause

    has performance ramifications for large tables.  Storage should be

    allocated to minimize dynamic allocation of additional space.

CLUSTER

    specifies that the table is to be part of the cluster.  The columns

    listed in this clause are the table columns that correspond to the

    cluster's columns.  Generally, the cluster columns of a table are

    the column or columns that comprise its primary key or a portion of

    its primary key.

    Specify one column from the table for each column in the cluster

    key.  The columns are matched by position, not by name.  Since a

    clustered table uses the cluster's space allocation, do not use the

    PCTFREE, PCTUSED, INITRANS, or MAXTRANS parameters, the TABLESPACE

    option, or the STORAGE clause in conjunction with the CLUSTER

    option.

ENABLE

    enables an integrity constraint.

DISABLE

    disables an integrity constraint.

    Constraints specified in the ENABLE and DISABLE clauses of a CREATE

    TABLE statement must be defined in the statement.  You can also

    enable and disable constraints with the ENABLE and DISABLE keywords

    of the CONSTRAINT clause.  If you define a constraint but do not

    explicitly enable or disable it, ORACLE enables it by default.

    You cannot use the ENABLE and DISABLE clauses in a CREATE TABLE

    statement to enable and disable triggers.

AS subquery

    inserts the rows returned by the subquery into the table upon its

    creation.

    If you include this clause, the column definitions can only specify

    column names, default values, and integrity constraints, not

    datatypes.  ORACLE derives column datatypes and lengths from the

    subquery.  ORACLE also automatically defines NOT NULL constraints on

    columns in the new table if they existed on the corresponding

    columns of the selected table and the subquery does not modify the

    column value with a SQL function or operator.  A CREATE TABLE

    statement cannot contain both the AS clause and a referential

    integrity constraint definition.

    The number of columns must equal the number of expressions in the

    subquery.  If all expressions in the subquery are columns, you can

    omit the columns from the table definition entirely.  In this case,

    the names of the columns of table are the same as the columns in the

    subquery.

PREREQUISITES:

    To create a table in your own schema, you must have CREATE TABLE

    system privilege.  To create a table in another user's schema, you

    must have CREATE ANY TABLE system privilege.  Also, the owner of the

    schema to contain the table must have either space quota on the

    tablespace to contain the table or UNLIMITED TABLESPACE system

    privilege.

SEE:

    ALTER TABLE, CONSTRAINT, CREATE CLUSTER, CREATE INDEX, CREATE

    TABLESPACE, DISABLE, DROP TABLE, ENABLE, STORAGE

    CONSTRAINT clause

PURPOSE:

    To define an integrity constraint.  An integrity constraint is a

    rule that restricts the values for one or more columns in a table.

SYNTAX:

Column constraint:

[CONSTRAINT constraint]

{ [NOT] NULL

| {UNIQUE | PRIMARY KEY}

|  REFERENCES [schema.]table [(column)]

        [ON DELETE CASCADE]

|  CHECK (condition) }

{ [ USING INDEX [PCTFREE integer]

                [INITRANS integer] [MAXTRANS integer]

                [TABLESPACE tablespace]

                [STORAGE storage\_clause] ]

  [ EXCEPTIONS INTO [schema.]table

| DISABLE }

Table constraint:

[CONSTRAINT constraint]

{ {UNIQUE | PRIMARY KEY} (column [,column] ...)

|  FOREIGN KEY (column [,column] ...)

         REFERENCES [schema.]table [(column [,column] ...)]

        [ON DELETE CASCADE]

| CHECK (condition) }

{ [ USING INDEX [PCTFREE integer]

                [INITRANS integer] [MAXTRANS integer]

                [TABLESPACE tablespace]

                [STORAGE storage\_clause] ]

  [ EXCEPTIONS INTO [schema.]table[@dblink]

| DISABLE }

where:

CONSTRAINT

    identifies the integrity constraint by the name constraint.  ORACLE

    stores this name in the data dictionary along with the definition of

    the integrity constraint.  If you omit this identifier, ORACLE

    generates a name with this form:

            SYS\_Cn

    where

            n

                   is an integer that makes the name unique

                   within the database.

    For the names and definitions of integrity constraints, query the

    data dictionary.

NULL

    specifies that a column can contain null values.

NOT NULL

    specifies that a column cannot contain null values.

    If you do not specify NULL or NOT NULL in a column definition, NULL

    is the default.

UNIQUE

    designates a column or combination of columns as a unique key.

PRIMARY KEY

    designates a column or combination of columns as the table's primary

    key.

FOREIGN KEY

    designates a column or combination of columns as the foreign key in

    a referential integrity constraint.

REFERENCES

    identifies the primary or unique key that is referenced by a foreign

    key in a referential integrity constraint.

ON DELETE CASCADE

    specifies that ORACLE maintains referential integrity by

    automatically removing dependent foreign key values if you remove a

    referenced primary or unique key value.

CHECK

    specifies a condition that each row in the table must satisfy.

USING INDEX

    specifies parameters for the index ORACLE uses to enforce a UNIQUE

    or PRIMARY KEY constraint.  The name of the index is the same as the

    name of the constraint.  You can choose the values of the INITRANS,

    MAXTRANS, TABLESPACE, STORAGE, and PCTFREE parameters for the index.

    For information on these parameters, see the CREATE TABLE command.

    Only use this clause when enabling UNIQUE and PRIMARY KEY

    constraints.

EXCEPTIONS INTO

    identifies a table into which ORACLE places information about rows

    that violate an enabled integrity constraint.  This table must exist

    before you use this option.  If you omit schema, ORACLE assumes the

    exception table is in your own schema.  The exception table must be

    on your local database.

DISABLE

    disables the integrity constraint.  If an integrity constraint is

    disabled, ORACLE does not enforce it.

    If you do not specify this option, ORACLE automatically enables the

    integrity constraint.

    You can also enable and disable integrity constraints with the

    ENABLE and DISABLE clauses of the CREATE TABLE and ALTER TABLE

    commands.

PREREQUISITES:

    CONSTRAINT clauses can appear in either CREATE TABLE or ALTER TABLE

    commands.  To define an integrity constraint, you must have the

    privileges necessary to issue one of these commands.  See the CREATE

    TABLE and ALTER TABLE commands.

    Defining a constraint may also require additional privileges or

    preconditions that depend on the type of constraint.

SEE:

    ALTER TABLE, CREATE TABLE, DISABLE, ENABLE

Example:  To define the table staff, as user scott, you could enter:

               CREATE TABLE staff (

                  empno    NUMBER       NOT NULL   PRIMARY KEY,

                  ename    CHAR(20)     NOT NULL   CHECK (ename = UPPER),

                  job      CHAR(10),

                  mgr      NUMBER       REFERENCES scott.staff(empno),

                  hiredate DATE         CHECK (hiredate >= SYSDATE - 7),

                  sal      NUMBER(10,2) CHECK (sal > 800),

                  comm     NUMBER(9,2)  DEFAULT NULL,

                  deptno   NOT NULL     REFERENCES scott.dept(deptno)

               )

               PCTFREE 5 PCTUSED 75