DBA Cheat Sheet

One note of caution: if you don’t know what a specific keyword of a command does, don’t use it without checking out its purpose. This is a reference for those who understand what something like cascade constraints means when associated with a drop table command.

alter cluster

```
ALTER CLUSTER pub_cluster SIZE 4K;
ALTER CLUSTER pub_cluster DEALLOCATE UNUSED KEEP 1M;
```

alter database: Alter a Data File

```
ALTER DATABASE DATAFILE 4 OFFLINE;
ALTER DATABASE DATAFILE '/opt/oracle/datafile/users01.dbf' OFFLINE;
ALTER DATABASE DATAFILE '/opt/oracle/datafile/users01.dbf'
RESIZE 100m;
ALTER DATABASE DATAFILE '/opt/oracle/datafile/users01.dbf'
AUTOEXTEND ON NEXT 100M MAXSIZE 1000M;
ALTER DATABASE DATAFILE 4 END BACKUP;
```

alter database: Alter a Tempfile

```
ALTER DATABASE TEMPFILE 4 RESIZE 100M;
ALTER DATABASE TEMPFILE 4
AUTOEXTEND ON NEXT 100M MAXSIZE 1000M;
ALTER DATABASE TEMPFILE 4 DROP INCLUDING DATAFILES;
ALTER DATABASE TEMPFILE 4 OFFLINE;
```

alter database: ARCHIVELOG Mode Commands

```
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE NOARCHIVELOG;
ALTER DATABASE FORCE LOGGING;
ALTER DATABASE CLEAR LOGFILE '/opt/oracle/logfiles/redo01.rdo';
ALTER DATABASE CLEAR UNARCHIVED LOGFILE
'/opt/oracle/logfiles/redo01.rdo';
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE);
ALTER DATABASE DROP SUPPLEMENTAL LOG DATA;
```

alter database: Control File Operations

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
ALTER DATABASE BACKUP CONTROLFILE TO TRACE
```

```
AS '/opt/oracle/logfile_backup/backup_logfile.trc'
REUSE RESETLOGS;
ALTER DATABASE BACKUP CONTROLFILE TO
'/opt/oracle/logfile_backup/backup_logfile.ctl';
```

alter database: Create a Data File

```
ALTER DATABASE CREATE DATAFILE
'/opt/oracle/datafile/users01.dbf' AS '/opt/oracle/datafile/users01.dbf';
ALTER DATABASE CREATE DATAFILE 4
AS '/opt/oracle/datafile/users01.dbf';
ALTER DATABASE CREATE DATAFILE
'/opt/oracle/datafile/users01.dbf' AS NEW;
```

alter database: Datafile Offline/Online

See alter database: Alter a Data File

alter database: Logfile Commands

```
ALTER DATABASE ADD LOGFILE GROUP 2
('/opt/oracle/logfiles/redo02a.rdo', '/opt/oracle/logfiles/redo02b.rdo')
SIZE 300M REUSE;
ALTER DATABASE ADD LOGFILE MEMBER
'/opt/oracle/logfiles/redo02c.rdo'
to GROUP 2;
ALTER DATABASE ADD LOGFILE thread 3 GROUP 2
('/opt/oracle/logfiles/redo02a.rdo', '/opt/oracle/logfiles/redo02b.rdo')
SIZE 300M REUSE;
ALTER DATABASE DROP LOGFILE GROUP 3;
ALTER DATABASE DROP LOGFILE MEMBER '/opt/oracle/logfiles/redo02b.rdo';
```

alter database: Mount and Open the Database

```
ALTER DATABASE MOUNT;
ALTER DATABASE OPEN;
```

alter database: Move or Rename a Database File or Online Redo Log

NOTE

The database must be mounted to rename or move online redo logs.
The database must be mounted, or the data files taken offline to move database data files.

```
ALTER DATABASE RENAME FILE '/ora/datafile/oldfile.dbf' TO
'/ora/datafile/newfile.dbf';
```

alter database: Open the Database Read-Only

```
ALTER DATABASE OPEN READ ONLY;
```

alter database: Open the Database with resetlogs

```
ALTER DATABASE OPEN RESETLOGS;
```

alter database: Recover the Database

For database recovery, I recommend the use of the recover command instead. See the “recover” section, later in the chapter.

alter function: Recompile a Function

```
ALTER FUNCTION my_function COMPILE;
```

alter index: Allocate and Deallocate Extents

```
ALTER INDEX ix_my_tab ALLOCATE EXTENT;
ALTER INDEX ix_my_tab ALLOCATE EXTENT
DATAFILE '/ora/datafile/newidx.dbf';
ALTER INDEX ix_my_tab DEALLOCATE UNUSED;
ALTER INDEX ix_my_tab DEALLOCATE UNUSED KEEP 100M;
```

alter index: Miscellaneous Maintenance

```
ALTER INDEX ix_my_tab PARALLEL 3;
ALTER INDEX ix_my_tab NOPARALLEL;
ALTER INDEX ix_my_tab NOCOMPRESS;
ALTER INDEX ix_my_tab COMPRESS;
```

alter index: Modify Logging Attributes

```
ALTER INDEX ix_my_tab LOGGING;
ALTER INDEX ix_my_tab NOLOGGING;
```

alter index: Modify Storage and Physical Attributes

```
ALTER INDEX ix_my_tab PCTFREE 10 PCTUSED 40 INITRANS 5
STORAGE (NEXT 100k MAXEXTENTS UNLIMITED FREELISTS 10
BUFFER_POOL KEEP);
```

alter index: Partition – Add Hash Index Partition

```
ALTER INDEX ix_my_tab ADD PARTITION
TABLESPACE NEWIDXTBS;
```

alter index: Partition – Coalesce Partition

```
ALTER INDEX ix_my_tab COALESCE PARTITION;
```

alter index: Partition – Drop Partition

```
ALTER INDEX ix_my_tab DROP PARTITION ix_my_tab_jan_04;
```

alter index: Partition – Modify Default Attributes

```
ALTER INDEX ix_my_tab MODIFY DEFAULT ATTRIBUTES
FOR PARTITION ix_my_tab_jan_04
PCTFREE 10 PCTUSED 40 TABLESPACE newidxtbs
NOLOGGING COMPRESS;
```

alter index: Partition – Modify Partition

```
ALTER INDEX ix_my_tab MODIFY PARTITION ix_my_tab_jan_04
DEALLOCATE UNUSED KEEP 100M;
ALTER INDEX ix_my_tab MODIFY PARTITION ix_my_tab_jan_04
ALLOCATE EXTENT SIZE 100m;
ALTER INDEX ix_my_tab MODIFY PARTITION ix_my_tab_jan_04
PCTUSED 40 STORAGE(NEXT 50m) NOLOGGING;
```

alter index: Partition – Modify Subpartition

```
ALTER INDEX ix_my_tab MODIFY SUBPARTITION ix_my_tab_jan_04
DEALLOCATE UNUSED KEEP 100M;
ALTER INDEX ix_my_tab MODIFY SUBPARTITION ix_my_tab_jan_04
ALLOCATE EXTENT SIZE 100m;
ALTER INDEX ix_my_tab MODIFY SUBPARTITION ix_my_tab_jan_04
PCTUSED 40 STORAGE(NEXT 50m) NOLOGGING;
```

alter index: Partition – Rename

```
ALTER INDEX ix_my_tab RENAME
PARTITION ix_my_tab_jan_04 TO ix_my_tab_jan_05;
ALTER INDEX ix_my_tab RENAME
SUBPARTITION ix_my_tab_jan_04 TO ix_my_tab_jan_05;
```

alter index: Partition – Split

```
ALTER INDEX ix_my_tab SPLIT PARTITION ix_my_tab_jan_05
AT ('15-JAN-05') INTO PARTITION ix_my_tab_jan_05a
TABLESPACE myidxtbs
STORAGE (INITIAL 100m NEXT 50M FREELISTS 5);
```

alter index: Rebuild Nonpartitioned Indexes

```
ALTER INDEX ix_my_tab REBUILD ONLINE;
ALTER INDEX ix_my_tab REBUILD ONLINE
TABLESPACE idx_tbs_new PCTFREE 1
STORAGE (INITIAL 50M NEXT 50m FREELISTS 5)
COMPUTE STATISTICS PARALLEL 0;
```

alter index: Rebuild Partitions

```
ALTER INDEX ix_my_tab
REBUILD PARTITION ix_my_tab_jan_04 ONLINE;
ALTER INDEX ix_my_tab
REBUILD SUBPARTITION ix_my_tab_jan_04 ONLINE
PCTFREE 1 STORAGE (INITIAL 50M NEXT 50m FREELISTS 5)
COMPUTE STATISTICS PARALLEL 0;
```

alter index: Rename

```
ALTER INDEX ix_my_tab RENAME TO 'ix_my_tab_01';
```

alter index: Shrink

```
ALTER INDEX ix_my_tab SHRINK SPACE;
ALTER INDEX ix_my_tab SHRINK SPACE COMPACT CASCADE;
```

alter materialized view: Allocate and Deallocate Extents

```
ALTER MATERIALIZED VIEW mv_my_tab ALLOCATE EXTENT;
ALTER MATERIALIZED VIEW mv_my_tab DEALLOCATE UNUSED;
```

alter materialized view: Miscellaneous

```
ALTER MATERIALIZED VIEW mv_my_tab COMPRESS;
ALTER MATERIALIZED VIEW mv_my_tab PARALLEL 3;
ALTER MATERIALIZED VIEW mv_my_tab NOLOGGING;
ALTER MATERIALIZED VIEW mv_my_tab LOGGING;
ALTER MATERIALIZED VIEW mv_my_tab CONSIDER FRESH;
ALTER MATERIALIZED VIEW mv_my_tab ENABLE QUERY REWRITE;
```

alter materialized view: Physical Attributes and Storage

```
ALTER MATERIALIZED VIEW mv_my_tab
PCTFREE 5 PCTUSED 60
STORAGE (NEXT 100m FREELISTS 5);
```

alter materialized view: Refresh

```
ALTER MATERIALIZED VIEW mv_my_tab REFRESH FAST;
ALTER MATERIALIZED VIEW mv_my_tab REFRESH COMPLETE;
ALTER MATERIALIZED VIEW mv_my_tab REFRESH FAST ON DEMAND;
ALTER MATERIALIZED VIEW mv_my_tab REFRESH FAST ON COMMIT;
ALTER MATERIALIZED VIEW mv_my_tab REFRESH COMPLETE
START WITH sysdate;
ALTER MATERIALIZED VIEW mv_my_tab REFRESH COMPLETE
START WITH sysdate NEXT sysdate+1/24;
```

alter materialized view: Shrink Space

```
ALTER MATERIALIZED VIEW mv_my_tab SHRINK SPACE;
ALTER MATERIALIZED VIEW mv_my_tab
SHRINK SPACE COMPACT CASCADE;
```

alter materialized view log: Add Components

```
ALTER MATERIALIZED VIEW LOG ON my_tab ADD PRIMARY KEY;
ALTER MATERIALIZED VIEW LOG ON my_tab ADD (col1, col2)
INCLUDING NEW VALUES;
ALTER MATERIALIZED VIEW LOG ON my_tab ADD (col1, col2),
ROWID, SEQUENCE INCLUDING NEW VALUES;
```

alter materialized view log: Allocate and Deallocate Extents

```
ALTER MATERIALIZED VIEW LOG ON my_tab ALLOCATE EXTENT;
ALTER MATERIALIZED VIEW LOG ON my_tab DEALLOCATE UNUSED;
```

alter materialized view log: Miscellaneous

```
ALTER MATERIALIZED VIEW LOG ON my_tab PARALLEL 3;
ALTER MATERIALIZED VIEW LOG ON my_tab NOLOGGING;
ALTER MATERIALIZED VIEW LOG ON my_tab SHRINK SPACE;
```

alter materialized view log: Physical Attributes and Storage

```
ALTER MATERIALIZED VIEW LOG ON my_tab
PCTFREE 5 PCTUSED 60
STORAGE (NEXT 100m FREELISTS 5);
```

alter package: Compile

```
ALTER PACKAGE pk_my_package COMPILE;
ALTER PACKAGE pk_my_package COMPILE SPECIFICATION;
ALTER PACKAGE pk_my_package COMPILE BODY;
```

alter procedure: Compile

```
ALTER PROCEDURE pk_my_package COMPILE;
```

alter profile: Miscellaneous

```
ALTER ROLE my_role IDENTIFIED BY password;
ALTER ROLE my_role NOT IDENTIFIED;
```

alter profile: Modify Limits (Password)

```
ALTER PROFILE my_profile LIMIT FAILED_LOGIN_ATTEMPTS=3;
ALTER PROFILE my_profile LIMIT PASSWORD_LOCK_TIME=2/24;
ALTER PROFILE my_profile LIMIT PASSWORD_GRACE_TIME=5;
ALTER PROFILE my_profile LIMIT PASSWORD_LIFETIME=60;
ALTER PROFILE my_profile LIMIT PASSWORD_REUSE_TIME=365
PASSWORD_REUSE_MAX=3;
```

alter profile: Modify Limits (Resource)

```
ALTER PROFILE my_profile LIMIT SESSIONS_PER_CPU=10;
ALTER PROFILE my_profile LIMIT CONNECT_TIME=1000;
ALTER PROFILE my_profile LIMIT IDLE_TIME=60;
ALTER PROFILE my_profile LIMIT PRIVATE_SGA=1000000;
```

alter rollback segment: Online/Offline

```
ALTER ROLLBACK SEGMENT rbs01 OFFLINE;
ALTER ROLLBACK SEGMENT rbs01 ONLINE;
```

alter rollback segment: Shrink

```
ALTER ROLLBACK SEGMENT rbs01 SHRINK;
ALTER ROLLBACK SEGMENT rbs01 SHRINK TO 100M;
```

alter rollback segment: storage Clause

```
ALTER ROLLBACK SEGMENT rbs01 STORAGE(NEXT 50M OPTIMAL 100M);
```

alter sequence: Miscellaneous

```
ALTER SEQUENCE my_seq INCREMENT BY -5;
ALTER SEQUENCE my_seq INCREMENT BY 1 MAXVALUE 50000 CYCLE;
ALTER SEQUENCE my_seq NOMAXVALUE;
ALTER SEQUENCE my_seq CACHE ORDER;
ALTER SEQUENCE my_seq INCREMENT BY 1
MINVALUE 1 MAXVALUE 500 CYCLE;
```

alter session: Enable and Disable Parallel Operations

```
ALTER SESSION ENABLE PARALLEL DML PARALLEL 3;
ALTER SESSION ENABLE PARALLEL DDL;
ALTER SESSION DISABLE PARALLEL QUERY;
```

alter session: Resumable Space Management

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600;
ALTER SESSION DISABLE RESUMABLE;
```

alter session: Set Session Parameters

```
ALTER SESSION SET nls_date_format='MM/DD/YYYY HH24:MI:SS';
ALTER SESSION SET sort_area_size=10000000;
ALTER SESSION SET query_rewrite_enabled=TRUE;
ALTER SESSION SET resumable_timeout=3600;
ALTER SESSION SET skip_unusable_indexes=TRUE;
ALTER SESSION SET SQL_TRACE=TRUE;
```

alter system: Logfile and Archive Logfile Management

```
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM ARCHIVE LOG START;
ALTER SYSTEM ARCHIVE LOG STOP;
ALTER SYSTEM ARCHIVE LOG ALL;
ALTER SYSTEM ARCHIVE LOG THREAD 1 ALL;
ALTER SYSTEM ARCHIVE LOG ALL TO 'C:\oracle\allarch';
```

alter system: Set System Parameters

```
ALTER SYSTEM SET db_cache_size=325M
COMMENT='This change is to add more memory to the system'
SCOPE=BOTH;
ALTER SYSTEM SET COMPATIBLE=10.0.0
COMMENT='GOING TO 10G!' SCOPE=SPFILE;
```

alter system: System Management

```
ALTER SYSTEM CHECKPOINT GLOBAL;
ALTER SYSTEM KILL SESSION '145,334';
ALTER SYSTEM ENABLE RESTRICTED SESSION;
ALTER SYSTEM DISABLE RESTRICTED SESSION;
ALTER SYSTEM SUSPEND;
ALTER SYSTEM QUIESCE RESTRICTED;
ALTER SYSTEM UNQUIESCE;
ALTER SYSTEM RESUME;
ALTER SYSTEM FLUSH SHARED_POOL;
ALTER SYSTEM FLUSH BUFFER_CACHE;
```

alter table: External Table Operations

```
ALTER TABLE ext_parts REJECT LIMIT 500;
ALTER TABLE ext_parts DEFUALT DIRECTORY ext_employee_dir;
ALTER TABLE ext_parts ACCESS PARAMETERS
(FIELDS TERMINATED BY ’,’);
ALTER TABLE ext_parts LOCATION (’PARTS01.TXT’,’PARTS02.TXT’);
ALTER TABLE ext_parts ADD COLUMN (SSN NUMBER);
```

alter table: Move Table

```
ALTER TABLE parts move TABLESPACE parts_new_tbs PCTFREE 10 PCTUSED 60;
```

alter table: Table Column – Add

```
ALTER TABLE PARTS ADD (part_location VARCHAR2(20) );
ALTER TABLE PARTS ADD (part_location VARCHAR2(20), part_bin VARCHAR2(30)
);
ALTER TABLE parts ADD (photo BLOB)
LOB (photo) STORE AS lob_parts_photo
(TABLESPACE parts_lob_tbs);
```

alter table: Table Column – Modify

```
ALTER TABLE PARTS MODIFY (part_location VARCHAR2(30) );
ALTER TABLE PARTS MODIFY
part_location VARCHAR2(30), part_bin VARCHAR2(20) );
ALTER TABLE parts modify (name NOT NULL);
ALTER TABLE parts modify (name NULL);
ALTER TABLE parts MODIFY LOB (photo) (STORAGE(FREELISTS 2));
ALTER TABLE parts MODIFY LOB (photo) (PCTVERSION 50);
```

alter table: Table Column – Remove

```
ALTER TABLE parts DROP (part_location);
ALTER TABLE parts DROP (part_location, part_bin);
```

alter table: Table Column – Rename

```
ALTER TABLE parts RENAME COLUMN part_location TO part_loc;
```

alter table: Table Constraints – Add Check Constraint

```
ALTER TABLE parts ADD (CONSTRAINT ck_parts_01 CHECK (id > 0) );
```

alter table: Table Constraints – Add Default Value

```
ALTER TABLE PARTS MODIFY (name DEFAULT 'Not Available');
ALTER TABLE PARTS ADD (vendor_code NUMBER DEFAULT 0);
ALTER TABLE PARTS MODIFY (part_description DEFAULT NULL);
```

alter table: Table Constraints – Add Foreign Key

```
ALTER TABLE parts ADD CONSTRAINT fk_part_bin
FOREIGN KEY (bin_code) REFERENCES part_bin;
```

alter table: Table Constraints – Add Primary and Unique Key

```
ALTER TABLE parts ADD CONSTRAINT pk_parts_part_id
PRIMARY KEY (id) USING INDEX TABLESPACE parts_index
STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0);
ALTER TABLE parts ADD CONSTRAINT uk_parts_part_bin
UNIQUE (part_bin)USING INDEX TABLESPACE parts_index
```

```
STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0);
```

alter table: Table Constraints – Modify

```
ALTER TABLE parts DISABLE UNIQUE (part_bin);
ALTER TABLE parts DISABLE CONSTRAINT uk_parts_part_bin;
ALTER TABLE parts DISABLE CONSTRAINT uk_parts_part_bin KEEP INDEX;
ALTER TABLE parts DISABLE CONSTRAINT fk_part_bin;
ALTER TABLE parts DISABLE CONSTRAINT fk_part_bin
DISABLE PRIMARY KEY KEEP INDEX;
ALTER TABLE parts ENABLE CONSTRAINT fk_part_bin;
ALTER TABLE parts ENABLE PRIMARY KEY;
ALTER TABLE parts ENABLE UNIQUE (part_bin);
ALTER TABLE parts ENABLE NOVALIDATE CONSTRAINT fk_part_bin;
ALTER TABLE parts ENABLE NOVALIDATE PRIMARY KEY;
ALTER TABLE parts ENABLE NOVALIDATE UNIQUE (part_bin);
ALTER TABLE parts ENABLE NOVALIDATE PRIMARY KEY
ENABLE NOVALIDATE CONSTRAINT fk_part_bin;
```

alter table: Table Constraints – Remove

```
ALTER TABLE parts DROP CONSTRAINT fk_part_bin;
ALTER TABLE parts DROP PRIMARY KEY;
ALTER TABLE parts DROP PRIMARY KEY CASCADE;
ALTER TABLE parts DROP UNIQUE (uk_parts_part_bin);
```

alter table: Table Partition – Add

```
ALTER TABLE store_sales ADD PARTITION sales_q1_04
VALUES LESS THAN (TO_DATE('01-APR-2004','DD-MON-YYYY'))
TABLESPACE data_0104_tbs UPDATE GLOBAL INDEXES;
ALTER TABLE daily_transactions ADD PARTITION;
ALTER TABLE daily_transactions
ADD PARTITION Alaska VALUES ('AK');
ALTER TABLE daily_transactions
add PARTITION SALES_2004_Q1 VALUES LESS THAN
(TO_DATE('01-APR-2004','DD-MON-YYYY')) SUBPARTITIONS 4;
```

alter table: Table Partition – Merge

```
ALTER TABLE store_sales
MERGE PARTITIONS Oklahoma, texas
INTO PARTITION oktx;
```

alter table: Table Partition – Move

```
ALTER TABLE store_sales MOVE PARTITION sales_overflow TABLESPACE
new_sales_overflow STORAGE (INITIAL 100m NEXT 100m PCTINCREASE 0)
UPDATE GLOBAL INDEXES;
```

alter table: Table Partition – Remove

```
ALTER TABLE store_sales DROP PARTITION sales_q1_04 UPDATE GLOBAL INDEXES;
```

alter table: Table Partition – Rename

```
ALTER TABLE store_sales RENAME PARTITION sales_q1 TO sales_first_quarter;
```

alter table: Table Partition – Split

```
ALTER TABLE store_sales
SPLIT PARTITION sales_overflow AT
(TO_DATE('01-FEB-2004','DD-MON-YYYY') )
INTO (PARTITION sales_q4_2003,
PARTITION sales_overflow)
UPDATE GLOBAL INDEXES;
ALTER TABLE composite_sales SPLIT PARTITION sales_q1
AT (TO_DATE('15-FEB-2003','DD-MON-YYYY'))
INTO (PARTITION sales_q1_01 SUBPARTITIONS 4
STORE IN (q1_01_tab1, q1_01_tab2, q1_01_tab3, q1_01_tab4),
PARTITION sales_q1_02 SUBPARTITIONS 4
STORE IN (q1_02_tab1, q1_02_tab2, q1_02_tab3, q1_02_tab4) )
UPDATE GLOBAL INDEXES;
```

alter table: Table Partition – Truncate

```
ALTER TABLE store_sales TRUNCATE PARTITION sales_overflow
UPDATE GLOBAL INDEXES;
```

alter table: Table Properties

```
ALTER TABLE parts PCTFREE 10 PCTUSED 60;
ALTER TABLE parts STORAGE (NEXT 1M);
ALTER TABLE parts PARALLEL 4;
```

alter table: Triggers – Modify Status

```
ALTER TABLE parts DISABLE ALL TRIGGERS;
ALTER TABLE parts ENABLE ALL TRIGGERS;
```

alter tablespace: Backups

```
ALTER TABLESPACE my_data_tbs BEGIN BACKUP;
ALTER TABLESPACE my_data_tbs END BACKUP;
```

alter tablespace: Data Files and Tempfiles

```
ALTER TABLESPACE mytbs
ADD DATAFILE '/ora100/oracle/mydb/mydb_mytbs_01.dbf' SIZE 100M;
```

```
ALTER TABLESPACE mytemp
ADD TEMPFILE '/ora100/oracle/mydb/mydb_mytemp_01.dbf'
SIZE 100M;
ALTER TABLESPACE mytemp AUTOEXTEND OFF;
ALTER TABLESPACE mytemp AUTOEXTEND ON NEXT 100m MAXSIZE 1G;
```

alter tablespace: Rename

```
ALTER TABLESPACE my_data_tbs RENAME TO my_newdata_tbs;
```

alter tablespace: Tablespace Management

```
ALTER TABLESPACE my_data_tbs DEFAULT
STORAGE (INITIAL 100m NEXT 100m FREELISTS 3);
ALTER TABLESPACE my_data_tbs MINIMUM EXTENT 500k;
ALTER TABLESPACE my_data_tbs RESIZE 100m;
ALTER TABLESPACE my_data_tbs COALESCE;
ALTER TABLESPACE my_data_tbs OFFLINE;
ALTER TABLESPACE my_data_tbs ONLINE;
ALTER TABLESPACE mytbs READ ONLY;
ALTER TABLESPACE mytbs READ WRITE;
ALTER TABLESPACE mytbs FORCE LOGGING;
ALTER TABLESPACE mytbs NOLOGGING;
ALTER TABLESPACE mytbs FLASHBACK ON;
ALTER TABLESPACE mytbs FLASHBACK OFF;
ALTER TABLESPACE mytbs RETENTION GUARANTEE;
ALTER TABLESPACE mytbs RETENTION NOGUARANTEE;
```

alter trigger

```
ALTER TRIGGER tr_my_trigger DISABLE;
ALTER TRIGGER tr_my_trigger ENABLE;
ALTER TRIGGER tr_my_trigger RENAME TO tr_new_my_trigger;
ALTER TRIGGER tr_my_trigger COMPILE;
```

alter user: Change Password

```
ALTER USER olduser IDENTIFIED BY newpassword;
ALTER USER olduser IDENTIFIED EXTERNALLY;
```

alter user: Password and Account Management

```
ALTER USER olduser PASSWORD EXPIRE;
ALTER USER olduser ACCOUNT LOCK;
ALTER USER olduser ACCOUNT UNLOCK;
```

alter user: Profile

```
ALTER USER olduser PROFILE admin_profile;
```

alter user: Quotas

```
ALTER USER olduser QUOTA UNLIMITED ON users;
ALTER USER olduser QUOTA 10000M ON USERS;
```

alter user: Roles

```
ALTER USER olduser DEFAULT ROLE admin_role;
ALTER USER olduser DEFAULT ROLE NONE;
ALTER USER olduser DEFAULT ROLE ALL EXCEPT admin_role;
```

alter user: Tablespace Assignments

```
ALTER USER olduser DEFAULT TABLESPACE users;
ALTER USER olduser TEMPORARY TABLESPACE temp;
```

alter view: Constraints

```
ALTER VIEW my_view
ADD CONSTRAINT u_my_view_01 UNIQUE (empno)
RELY DISABLE NOVALIDATE;
ALTER VIEW my_view DROP CONSTRAINT u_my_view_01;
ALTER VIEW my_view DROP PRIMARY KEY;
ALTER VIEW my_view MODIFY CONSTRAINT u_my_view_01 NORELY;
ALTER VIEW my_view MODIFY CONSTRAINT u_my_view_01 RELY;
```

alter view: Recompile

```
ALTER VIEW my_view RECOMPILE;
```

analyze: Analyze Cluster

```
ANALYZE CLUSTER my_cluster_tab COMPUTE STATISTICS FOR ALL ROWS;
ANALYZE CLUSTER my_cluster_tab
ESTIMATE STATISTICS SAMPLE 10000 ROWS FOR ALL ROWS;
```

analyze: Analyze Index

```
ANALYZE INDEX ix_tab_01 COMPUTE STATISTICS FOR ALL ROWS;
ANALYZE INDEX ix_tab_01
ESTIMATE STATISTICS SAMPLE 10000 ROWS FOR ALL ROWS;
```

analyze: Analyze Table

```
ANALYZE TABLE mytab COMPUTE STATISTICS
FOR ALL INDEXED COLUMNS SIZE 100;
ANALYZE TABLE mytab COMPUTE STATISTICS
FOR ALL INDEXES;
```

audit

```
AUDIT ALL ON scott.emp;
AUDIT UPDATE, DELETE ON scott.emp;
AUDIT SELECT on scott.emp WHENEVER NOT SUCCESSFUL;
AUDIT INSERT, UPDATE, DELETE ON DEFAULT;
```

comment

```
COMMENT ON TABLE scott.mytab IS
'This is a comment on the mytab table';
COMMENT ON COLUMN scott.mytab.col1 IS
'This is a comment on the col1 column';
COMMENT ON MATERIALIZED VIEW scott.mview IS
'This is a comment on the materialized view mview';
```

create cluster

```
CREATE CLUSTER pub_cluster (pubnum NUMBER)
SIZE 8K PCTFREE 10 PCTUSED 60 TABLESPACE user_data;
CREATE CLUSTER pub_cluster (pubnum NUMBER)
SIZE 8K HASHKEYS 1000 PCTFREE 10 PCTUSED 60
TABLESPACE user_data;
```

create control file

```
CREATE CONTROLFILE REUSE DATABASE "mydb"
NORESETLOGS NOARCHIVELOG
MAXLOGFILES 32 MAXLOGMEMBERS 3
MAXDATAFILES 200 MAXINSTANCES 1
MAXLOGHISTORY 1000
LOGFILE
GROUP 1 ('/ora01/oracle/mydb/mydb_redo1a.rdo',
'/ora02/oracle/mydb/mydb_redo1b.rdo') SIZE 500K,
GROUP 2 ('/ora01/oracle/mydb/mydb_redo2a.rdo',
'/ora01/oracle/mydb/mydb_redo2b.rdo') SIZE 500K
DATAFILE
'/ora01/oracle/mydb/mydb_system_01.dbf ',
'/ora01/oracle/mydb/mydb_users_01.dbf ',
'/ora01/oracle/mydb/mydb_undo_01.dbf ',
'/ora01/oracle/mydb/mydb_sysaux_01.dbf ',
'/ora01/oracle/mydb/mydb_alldata_01.dbf ';
```

create database

```
CREATE DATABASE prodb
MAXINSTANCES 1 MAXLOGHISTORY 1
MAXLOGFILES 5 MAXLOGMEMBERS 3
MAXDATAFILES 100
DATAFILE 'C:\oracle\ora92010\prodb\system01.dbf'
SIZE 250M REUSE AUTOEXTEND ON NEXT 10240K
MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL DEFAULT
TEMPORARY TABLESPACE TEMP
TEMPFILE 'C:\oracle\ora92010\prodb\temp01.dbf'
SIZE 40M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
SYSAUX TABLESPACE
DATAFILE 'C:\oracle\ora92010\prodb\sysauxtbs01.dbf'
SIZE 300M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
UNDO TABLESPACE "UNDOTBS1"
DATAFILE 'C:\oracle\ora92010\prodb\undotbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
CHARACTER SET WE8MSWIN1252
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
```

```
GROUP 1 ('C:\oracle\ora92010\prodb\redo01.log') SIZE 102400K,
GROUP 2 ('C:\oracle\ora92010\prodb\redo02.log') SIZE 102400K,
GROUP 3 ('C:\oracle\ora92010\prodb\redo03.log') SIZE 102400K;
```

create database link

```
CREATE DATABASE LINK my_db_link
CONNECT TO current_user
USING 'my_db';
CREATE PUBLIC DATABASE LINK my_db_link
CONNECT TO remote_user IDENTIFIED BY psicorp
USING 'my_db';
```

create directory

```
CREATE OR REPLACE DIRECTORY mydir AS
'/opt/oracle/admin/directories/mydir';
```

create function

```
CREATE OR REPLACE FUNCTION find_value_in_table
(p_value IN NUMBER, p_table IN VARCHAR2,
p_column IN VARCHAR2)
RETURN NUMBER IS
v_found NUMBER;
v_sql VARCHAR2(2000);
BEGIN
v_sql:='SELECT 1 FROM '||p_table||' WHERE '||p_column||
' = '||p_value;
execute immediate v_sql into v_found;
return v_found;
END;
/
```

create index: Function-Based Index

```
CREATE INDEX fb_upper_last_name_emp ON emp_info (UPPER(last_name) );
```

create index: Global Partitioned Indexes

```
CREATE INDEX ix_part_my_tab_01 ON store_sales (invoice_number)
GLOBAL PARTITION BY RANGE (invoice_number)
(PARTITION part_001 VALUES LESS THAN (1000),
PARTITION part_002 VALUES LESS THAN (10000),
PARTITION part_003 VALUES LESS THAN (MAXVALUE) );
CREATE INDEX ix_part_my_tab_02 ON store_sales
(store_id, time_id)
GLOBAL PARTITION BY RANGE (store_id, time_id)
(PARTITION PART_001 VALUES LESS THAN
(1000, TO_DATE('04-01-2003','MM-DD-YYYY') )
TABLESPACE partition_001
STORAGE (INITIAL 100M NEXT 200M PCTINCREASE 0),
PARTITION part_002 VALUES LESS THAN
(1000, TO_DATE('07-01-2003','MM-DD-YYYY') )
TABLESPACE partition_002
STORAGE (INITIAL 200M NEXT 400M PCTINCREASE 0),
PARTITION part_003 VALUES LESS THAN (maxvalue, maxvalue)
TABLESPACE partition_003 );
```

create index: Local Partitioned Indexes

```
CREATE INDEX ix_part_my_tab_01 ON my_tab
(col_one, col_two, col_three)
LOCAL (PARTITION tbs_part_01 TABLESPACE part_tbs_01,
PARTITION tbs_part_02 TABLESPACE part_tbs_02,
PARTITION tbs_part_03 TABLESPACE part_tbs_03,
PARTITION tbs_part_04 TABLESPACE part_tbs_04);
CREATE INDEX ix_part_my_tab_01 ON my_tab (col_one, col_two, col_three)
LOCAL STORE IN (part_tbs_01, part_tbs_02, part_tbs_03, part_tbs_04);
CREATE INDEX ix_part_my_tab_01 ON my_tab (col_one, col_two, col_three)
LOCAL STORE IN (
part_tbs_01 STORAGE (INITIAL 10M NEXT 10M MAXEXTENTS 200),
part_tbs_02,
part_tbs_03 STORAGE (INITIAL 100M NEXT 100M MAXEXTENTS 200),
part_tbs_04 STORAGE (INITIAL 1000M NEXT 1000M MAXEXTENTS 200));
```

create index: Local Subpartitioned Indexes

```
CREATE INDEX sales_ix ON store_sales(time_id, store_id)
STORAGE (INITIAL 1M MAXEXTENTS UNLIMITED) LOCAL
(PARTITION q1_2003,
PARTITION q2_2003,
PARTITION q3_2003
(SUBPARTITION pq3200301, SUBPARTITION pq3200302,
SUBPARTITION pq3200303, SUBPARTITION pq3200304,
SUBPARTITION pq3200305),
PARTITION q4_2003
(SUBPARTITION pq4200301 TABLESPACE tbs_1,
SUBPARTITION pq4200302 TABLESPACE tbs_1,
SUBPARTITION pq4200303 TABLESPACE tbs_1,
SUBPARTITION pq4200304 TABLESPACE tbs_1,
SUBPARTITION pq4200305 TABLESPACE tbs_1,
SUBPARTITION pq4200306 TABLESPACE tbs_1,
SUBPARTITION pq4200307 TABLESPACE tbs_1,
SUBPARTITION pq4200308 TABLESPACE tbs_1),
PARTITION sales_overflow
(SUBPARTITION pqoflw01 TABLESPACE tbs_2,
SUBPARTITION pqoflw02 TABLESPACE tbs_2,
SUBPARTITION pqoflw03 TABLESPACE tbs_2,
SUBPARTITION pqoflw04 TABLESPACE tbs_2));
```

create index: Nonpartitioned Indexes

```
CREATE INDEX ix_mytab_01 ON mytab(column_1);
CREATE UNIQUE INDEX ix_mytab_01 ON mytab(column_1, column_2, column_3);
CREATE INDEX ix_mytab_01 ON mytab(column_1, column_2, column_3)
```

```
TABLESPACE my_indexes COMPRESS
STORAGE (INITIAL 10K NEXT 10K PCTFREE 10) COMPUTE STATISTICS;
CREATE BITMAP INDEX bit_mytab_01 ON my_tab(col_two)
TABLESPACE my_tbs;
```

create materialized view

```
CREATE MATERIALIZED VIEW emp_dept_mv1
TABLESPACE users BUILD IMMEDIATE
REFRESH FAST ON COMMIT WITH ROWID
ENABLE QUERY REWRITE AS
SELECT d.rowid deptrowid, e.rowid emprowid,
e.empno, e.ename, e.job, d.loc
FROM dept d, emp e
WHERE d.deptno = e.deptno;
CREATE MATERIALIZED VIEW emp_dept_mv3
TABLESPACE users BUILD IMMEDIATE
REFRESH FAST ON COMMIT WITH ROWID
DISABLE QUERY REWRITE AS
SELECT d.rowid deptrowid, e.rowid emprowid,
d.dname, d.loc, e.ename, e.job
FROM dept d, emp e
WHERE d.deptno (+) = e.deptno;
```

create materialized view: Partitioned Materialized View

```
CREATE MATERIALIZED VIEW part_emp_mv1
PARTITION BY RANGE (hiredate)
(PARTITION month1
VALUES LESS THAN (TO_DATE('01-APR-1981', 'DD-MON-YYYY'))
PCTFREE 0 PCTUSED 99
STORAGE (INITIAL 64k NEXT 16k PCTINCREASE 0)
TABLESPACE users,
```

create procedure

```
CREATE OR REPLACE PROCEDURE new_emp_salary
(p_empid IN NUMBER, p_increase IN NUMBER)
AS
BEGIN
UPDATE emp SET salary=salary*p_increase
WHERE empid=p_empid;
END;
/
```

create profile

```
CREATE PROFILE development_profile
LIMIT
SESSIONS_PER_USER 2 CONNECT_TIME 100000 IDLE_TIME 100000
LOGICAL_READS_PER_SESSION 1000000
PRIVATE_SGA 10m
```

```
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LIFE_TIME 60
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX 3
PASSWORD_LOCK_TIME 30
PASSWORD_GRACE_TIME 5;
```

create role

```
CREATE ROLE developer_role IDENTIFIED USING develop;
```

create rollback segment

```
CREATE ROLLBACK SEGMENT r01 TABLESPACE RBS
STORAGE (INITIAL 100m NEXT 100M MINEXTENTS 5 OPTIMAL 500M);
```

create sequence

```
CREATE SEQUENCE my_seq
START WITH 1 INCREMENT BY 1 MAXVALUE 1000000 CYCLE CACHE;
```

create spfile

```
CREATE SPFILE FROM PFILE;
CREATE SPFILE='/opt/oracle/admin/mydb/pfile/spfilemybd.ora'
FROM PFILE='/opt/oracle/admin/mydb/pfile/initmybd.ora';
```

create synonym

```
CREATE SYNONYM scott_user.emp FOR scott.EMP;
CREATE PUBLIC SYNONYM emp FOR scott.EMP;
```

create procedure

```
CREATE OR REPLACE PROCEDURE new_emp_salary
(p_empid IN NUMBER, p_increase IN NUMBER)
AS
BEGIN
UPDATE emp SET salary=salary*p_increase
WHERE empid=p_empid;
END;
/
```

create profile

```
CREATE PROFILE development_profile
LIMIT
SESSIONS_PER_USER 2 CONNECT_TIME 100000 IDLE_TIME 100000
LOGICAL_READS_PER_SESSION 1000000
PRIVATE_SGA 10m
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LIFE_TIME 60
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX 3
```

```
PASSWORD_LOCK_TIME 30
PASSWORD_GRACE_TIME 5;
```

create role

```
CREATE ROLE developer_role IDENTIFIED USING develop;
```

create rollback segment

```
CREATE ROLLBACK SEGMENT r01 TABLESPACE RBS
STORAGE (INITIAL 100m NEXT 100M MINEXTENTS 5 OPTIMAL 500M);
```

create sequence

```
CREATE SEQUENCE my_seq
START WITH 1 INCREMENT BY 1 MAXVALUE 1000000 CYCLE CACHE;
```

create spfile

```
CREATE SPFILE FROM PFILE;
CREATE SPFILE='/opt/oracle/admin/mydb/pfile/spfilemybd.ora'
FROM PFILE='/opt/oracle/admin/mydb/pfile/initmybd.ora';
```

create synonym

```
CREATE SYNONYM scott_user.emp FOR scott.EMP;
CREATE PUBLIC SYNONYM emp FOR scott.EMP;
```

create table

```
CREATE TABLE my_tab
(id NUMBER, current_value VARCHAR2(2000) ) COMPRESS;
CREATE TABLE parts (id NUMBER, version NUMBER, name VARCHAR2(30),
Bin_code NUMBER, upc NUMBER, active_code VARCHAR2(1) NOT NULL
CONSTRAINT ck_parts_active_code_01
CHECK (UPPER(active_code)= 'Y' or UPPER(active_code)='N'),
CONSTRAINT pk_parts PRIMARY KEY (id, version)
USING INDEX TABLESPACE parts_index
STORAGE (INITIAL 1m NEXT 1m) )
TABLESPACE parts_tablespace
PCTFREE 20 PCTUSED 60 STORAGE ( INITIAL 10m NEXT 10m PCTINCREASE 0);
```

create tablespace: Permanent Tablespace

```
CREATE TABLESPACE data_tbs
DATAFILE '/opt/oracle/mydbs/data/mydbs_data_tbs_01.dbf'
SIZE 100m;
CREATE TABLESPACE data_tbs
DATAFILE '/opt/oracle/mydbs/data/mydbs_data_tbs_01.dbf'
SIZE 100m FORCE LOGGING BLOCKSIZE 8k;
CREATE TABLESPACE data_tbs
DATAFILE '/opt/oracle/mydbs/data/mydbs_data_tbs_01.dbf'
SIZE 100m NOLOGGING
DEFAULT COMPRESS EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
CREATE TABLESPACE data_tbs
DATAFILE '/opt/oracle/mydbs/data/mydbs_data_tbs_01.dbf'
SIZE 100m NOLOGGING
DEFAULT COMPRESS EXTENT MANAGEMENT LOCAL AUTOALLOCATE
```

```
SEGMENT SPACE MANAGEMENT AUTO;
CREATE BIGFILE TABLESPACE data_tbs
DATAFILE '/opt/oracle/mydbs/data/mydbs_data_tbs_01.dbf'
SIZE 10G;
```

create tablespace: Temporary Tablespace

```
CREATE TABLESPACE temp_tbs
TEMPFILE '/opt/oracle/mydbs/data/mydbs_temp_tbs_01.tmp'
SIZE 100m;
```

create tablespace: Undo Tablespace

```
CREATE TABLESPACE undo_tbs
TEMPFILE '/opt/oracle/mydbs/data/mydbs_undo_tbs_01.tmp'
SIZE 1g RETENTION GUARANTEE;
```

create trigger

```
CREATE OR REPLACE TRIGGER emp_comm_after_insert
BEFORE INSERT ON emp FOR EACH ROW
DECLARE
v_sal number;
v_comm number;
BEGIN
-- Find username of person performing the INSERT into the table
v_sal:=:new.salary;
:new.comm:=v_sal*.10;
END;
/
```

create user

```
CREATE USER Robert IDENTIFIED BY Freeman
DEFAULT TABLESPACE users_tbs
TEMPORARY TABLESPACE temp
QUOTA 100M ON users_tbs
QUOTA UNLIMITED ON data_tbs;
```

create view

```
CREATE OR REPLACE VIEW vw_emp_dept_10 AS
SELECT * FROM EMP WHERE dept=10;
CREATE OR REPLACE VIEW vw_public_email AS
SELECT ename_first, ename_last, email_address
FROM EMP WHERE public='Y'
```

delete

```
DELETE FROM emp WHERE empid=100;
DELETE FROM emp e WHERE e.rowid >
(SELECT MIN (esub.ROWID) FROM emp esub
WHERE e.empid=esub.empid);
```

drop cluster

```
DROP CLUSTER scott.emp_cluster
INCLUDING TABLES CASCADE CONSTRAINTS;
```

drop database

```
DROP DATABASE;
```

drop database link

```
DROP DATABASE LINK my_db_link;
DROP PUBLIC DATABASE LINK my_db_link;
```

drop directory

```
DROP DIRECTORY mydir;
```

drop function

```
DROP FUNCTION find_value_in_table;
```

drop index

```
DROP INDEX ix_my_tab;
```

drop materialized view

```
DROP MATERIALIZED VIEW my_mview;
DROP MATERIALIZED VIEW my_mview PRESERVE TABLE;
```

drop materialized view log

```
DROP MATERIALIZED VIEW LOG ON mytab;
```

drop package/drop package body

```
DROP PACKAGE scott.my_package
DROP PACKAGE BODY scott.my_package;
```

drop procedure

```
DROP PROCEDURE my_proc;
```

drop profile

```
DROP PROFILE my_profile CASCADE;
```

drop role

```
DROP ROLE my_role;
```

drop rollback segment

```
DROP ROLLBACK SEGMENT rbs01;
```

drop sequence

```
DROP SEQUENCE my_seq;
```

drop synonym

```
DROP SYNONYM my_synonym;
DROP PUBLIC SYNONYM my_synonym;
```

drop table

```
DROP TABLE my_tab;
DROP TABLE my_tab CASCADE CONSTRAINTS;
DROP TABLE my_tab CASCADE CONSTRAINTS PURGE;
```

drop tablespace

```
DROP TABLESPACE my_tbs;
DROP TABLESPACE my_tbs INCLUDING CONTENTS;
DROP TABLESPACE my_tbs INCLUDING CONTENTS
AND DATAFILES CASCADE CONSTRAINTS;
```

drop trigger

```
DROP TRIGGER my_trigger;
```

drop user

```
DROP USER my_user CASCADE;
```

drop view

```
DROP VIEW my_view CASCADE CONSTRAINTS;
```

explain plan

```
EXPLAIN PLAN SET STATEMENT_ID='TEST' FOR
SELECT * FROM emp WHERE EMPID=100;
```

flashback database

```
FLASHBACK DATABASE TO SCN 10000;
FLASHBACK DATABASE TO TIMESTAMP SYSDATE – 1/24;
FLASHBACK DATABASE TO BEFORE TIMESTAMP SYSDATE – 1/24;
```

flashback table

```
FLASHBACK TABLE my_tab TO SCN 10000;
FLASHBACK TABLE my_tab TO TIMESTAMP SYSDATE – 1/24
ENABLE TRIGGERS;
FLASHBACK TABLE my_tab TO BEFORE DROP;
FLASHBACK TABLE my_tab TO BEFORE DROP RENAME TO rec_tab;
```

grants: Object Grants

```
GRANT SELECT ON scott.my_tab TO my_user;
GRANT INSERT, UPDATE, SELECT ON scott.my_tab TO my_user;
GRANT SELECT ON scott.my_tab TO my_user WITH GRANT OPTION;
GRANT SELECT ON scott.my_tab TO PUBLIC WITH GRANT OPTION;
```

grants: System Grants

```
GRANT CREATE TABLE to my_user;
GRANT CREATE ANY TABLE to my_user WITH ADMIN OPTION;
GRANT ALL PRIVILEGES to my_user WITH ADMIN OPTION;
```

insert

```
INSERT INTO dept VALUES (100, 'Marketing', 'Y');
INSERT INTO dept (deptid, dept_name, active)
VALUES (100, 'Marketing', 'Y');
INSERT INTO emp_history SELECT * FROM emp a
WHERE a.empid NOT IN (SELECT empid FROM emp_history);
INSERT INTO emp_pay_summary
SELECT empid, sum(gross_pay) FROM emp_pay_history
GROUP BY empid;
INSERT ALL
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date, deptid, mon_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date+1, deptid, tue_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date+2, deptid, wed_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date+3, deptid, thur_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date+4, deptid, fri_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
VALUES (store_id, start_date+5, deptid, sat_sales)
INTO store_sales (store_id, sales_date, deptid, sales_amt)
```

```
VALUES (store_id, start_date+6, deptid, sun_sales)
SELECT store_id, start_date, deptid, mon_sales, tue_sales,
wed_sales, thur_sales, fri_sales, sat_sales, sun_sales
FROM store_sales_load;
INSERT ALL
WHEN store_id < 100 THEN INTO east_stores
WHEN store_id >= 100 THEN INTO west_stores
ELSE INTO misc_stores
SELECT * FROM store_sales_load;
INSERT /*+ APPEND */ INTO emp VALUES (100,
'Jacob','Freeman',1000,20, null, 10, sysdate, 100,
sysdate+365);
```

lock table

```
LOCK TABLE my_table IN EXCLUSIVE MODE NOWAIT;
LOCK TABLE my_table IN ROW EXCLUSIVE MODE;
```

Merge

```
MERGE INTO emp_retire A
USING (SELECT empno, ename_last, ename_first, salary
FROM emp WHERE retire_cd='Y') B
ON (a.empid=b.empid)
WHEN MATCHED THEN UPDATE SET
a.ename_last=b.ename_last,
a.ename_first=b.ename_first,
a.salary=b.salary
DELETE WHERE (b.retire_cd='D')
WHEN NOT MATCHED THEN INSERT
(a.empid, a.ename_last, a.ename_first, a.salary)
VALUES (b.empid, b.ename_last, b.ename_first, b.salary)
WHERE (b.retire_cd!='D');
```

noaudit

```
NOAUDIT ALL ON scott.emp;
NOAUDIT UPDATE, DELETE ON scott.emp;
NOAUDIT SELECT on scott.emp WHENEVER NOT SUCCESSFUL;
NOAUDIT INSERT, UPDATE, DELETE ON DEFAULT;
```

purge

```
PURGE TABLE my_tab;
PURGE INDEX ix_my_tab;
PURGE RECYCLEBIN;
PURGE DBA_RECYCLEBIN;
PURGE TABLESPACE data_tbs USER scott;
```

recover

```
RECOVER DATABASE;
RECOVER TABLESPACE user_data, user_index;
RECOVER DATAFILE
'/opt/oracle/admin/mydb/datafile/mydb_users_01.dbf';
RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;
RECOVER DATABASE UNTIL CHANGE 94044;
RECOVER DATABASE UNTIL TIME '2004-08-01:22:00:04';
```

rename

```
RENAME my_table to my_tab;
```

revoke: Object Grants

```
REVOKE SELECT ON scott.my_tab FROM my_user;
REVOKE INSERT, UPDATE, SELECT ON scott.my_tab FROM my_user;
REVOKE SELECT ON scott.my_tab FROM my_user;
REVOKE SELECT ON scott.my_tab FROM PUBLIC;
```

revoke: System Grants

```
REVOKE CREATE TABLE FROM my_user;
REVOKE CREATE ANY TABLE FROM my_user;
REVOKE ALL PRIVILEGES FROM my_user;
```

rollback

```
ROLLBACK;
```

savepoint

```
SAVEPOINT alpha;
```

select

```
SELECT ename_last, dname
FROM emp a, dept b
WHERE a.deptid=b.deptid;
SELECT a.empid, b.dept_name
FROM emp a, dept b
WHERE a.deptid=b.deptid (+);
SELECT a.empid, b.dept_name
FROM emp a LEFT OUTER JOIN dept b
ON a.deptid=b.deptid;
SELECT * FROM dept WHERE EXISTS
(SELECT * FROM emp
WHERE emp.deptid=dept.deptid
AND emp.salary > 100);
SELECT ename_first, ename_last,
CASE deptid
```

```
WHEN 10 THEN 'Acounting' WHEN 20 THEN 'Sales'
ELSE 'None' END FROM emp;
SELECT empid, ename_last, salary, comm
FROM emp a
WHERE salary*.10 > (SELECT AVG(comm) FROM emp z
WHERE a.deptid=z.deptid);
WITH avg_dept_sales AS (
SELECT a.deptid, avg(b.sales_amt) avg_sales
FROM emp a, dept_sales b
WHERE a.deptid=b.deptid
GROUP BY a.deptid),
emp_salaries AS
(SELECT empid, AVG(salary) avg_salary FROM emp
GROUP BY empid)
SELECT * FROM emp_salaries b WHERE avg_salary*.05 >
(SELECT avg_sales FROM avg_dept_sales);
SELECT /*+ INDEX (a, emp_last_name_ix) */ empid
FROM emp a WHERE ename_last='Freeman'
SELECT empid, TO_CHAR(retire_date, 'MM/DD/YYYY')
FROM emp
Color profile: Generic CMYK printer profile
Composite Default screen
WHERE retire_date IS NOT NULL
ORDER BY retire_date
SELECT empid, COUNT(*)
FROM emp
GROUP BY empid
HAVING COUNT(*) > 1;
SELECT empid, salary FROM emp
AS OF TIMESTAMP(SYSTIMESTAMP - INTERVAL '1' DAY)
WHERE empid=20;
SELECT empid, salary FROM emp
VERSIONS BETWEEN
TIMESTAMP SYSTIMESTAMP - INTERVAL '1' DAY AND
SYSTIMESTAMP - INTERVAL '1' HOUR
WHERE empid=20;
```

set constraints

```
SET CONSTRAINTS ALL IMMEDIATE;
SET CONSTRAINTS ALL DEFERRED;
SET CONSTRAINT fk_my_tab DEFERRED;
```

set transaction

```
SET TRANSACTION USE ROLLBACK SEGMENT rbs01;
SET TRANSACTION READ ONLY;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

truncate

```
TRUNCATE TABLE my_tab;
TRUNCATE TABLE my_tab PRESERVE MATERIALIZED VIEW LOG;
TRUNCATE TABLE my_tab REUSE STORAGE;
TRUNCATE TABLE my_tab DROP STORAGE;
```

update

```
UPDATE emp SET salary=100 WHERE empid=100;
UPDATE emp SET salary=NULL, retire_date=SYSDATE
WHERE empid=100;
UPDATE emp SET salary=salary*1.10
WHERE deptid IN
(SELECT deptid FROM dept WHERE dept_name = 'Sales');
UPDATE emp a SET (salary, comm)=
(SELECT salary*1.10, comm*1.10
FROM emp b WHERE a.empid=b.empid);
INSERT INTO store_sales
PARTITION (store_sales_jan_2004) sa
SET sa.sales_amt=1.10 where store_id=100;
```