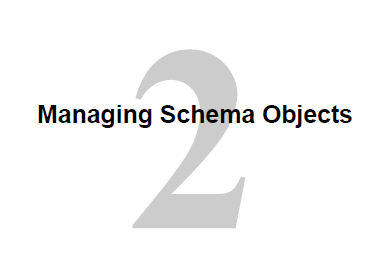
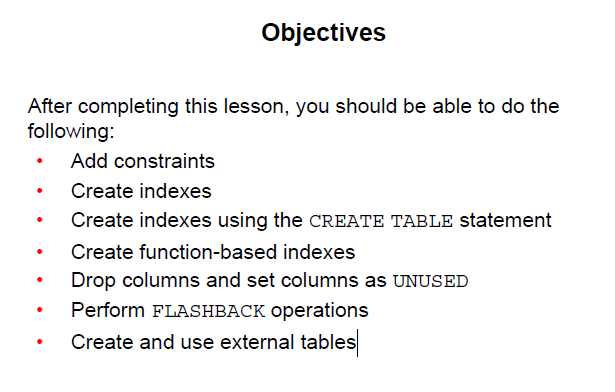
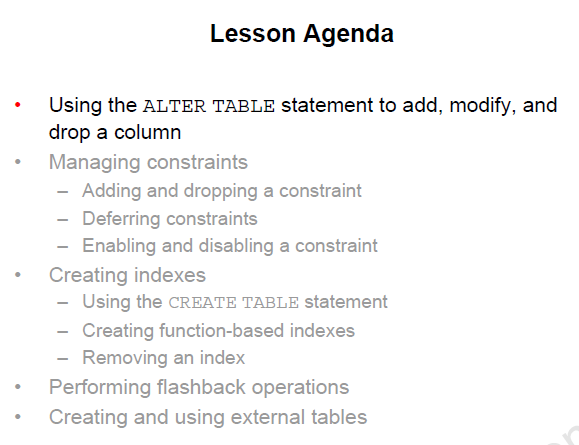
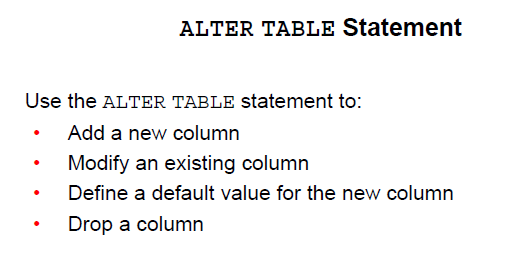
Les11 - ALTER

From chapter 2 of part 2









You may need to make changes to a table you created.

Also businesses will change over time

Alter Table Statement Syntax

SYNTAX

ALTER

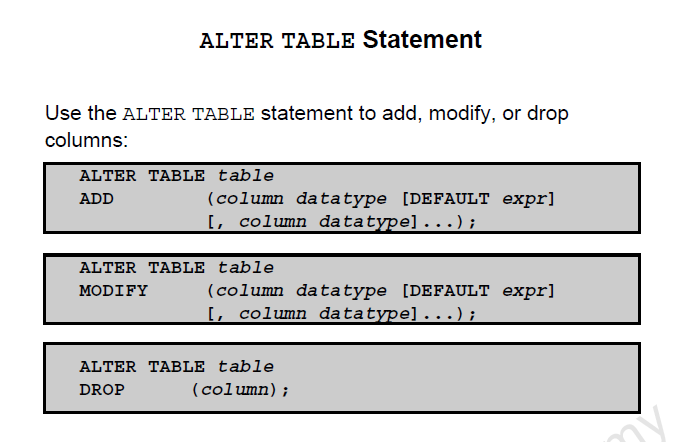
TABLE - name of the table

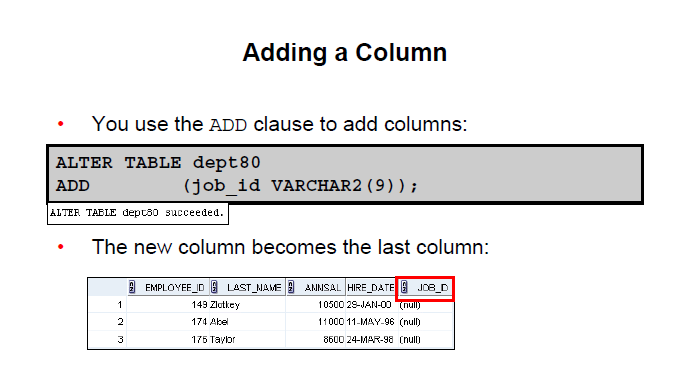
ADD – MODIFY – DROP is the type of modification

COLUMN -- name of column effected

DATATYPE -- datatype and length of the column

DEFAULT expr – specifies the default value for a column





You can add or modify a column

You cannot determine the order of the columns

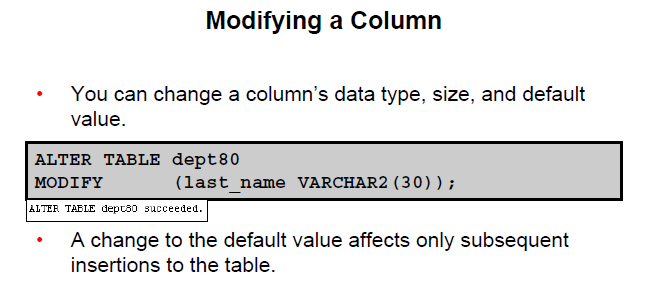
In the above, the column added is the last column

**Things to note:**

**If table has row of values, then the column is either null or the default**

**If column is NOT NULL, then it needs a default value to start with**

**If table is empty you can add a NOT NULL without the need for a default value.**



GUIDELINES:

Increase width or precision of numeric columns

Increase width of character columns

You can decrease width if

- The column has only NULL values

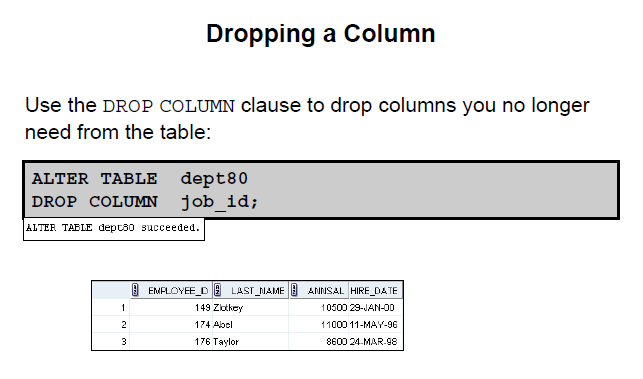
- The table has no rows

- The decrease is not less than the values in the column

You can change the data type if null

- except CHAR to VARCHAR is allowed

A change to the default only effects all late inserts



**Only 1 column dropped at a time**

The column can or cannot have data in it

After the alter there must be at least 1 column left in the table

Columns that are part of a constraint cannot be dropped

Set UNUSED option

Marks 1 or more columns as unused.

Dropping may be slow if there is a lot of data to drop

May be better to mark as unused and drop later when fewer users on system

SELECT \* will not show the data even though it is still there

DESCRIBE will not show the columns

You can DROP UNUSED COLUMN to remove a column marked as unused.

Sample:

**ALTER TABLE DEPARTMENTS**

**SET UNUSED (PHONE);** 🡸 message results in ALTER TABLE succeeded

This removed access to the data

**ALTER TABLE DEPARTMENTS**

**DROP UNUSED COLUMNS;** 🡸 message results in ALTER TABLE succeeded

This removes the data and frees up space

ALTER TABLE – ADD CONSTRAINTS

ADD constraints

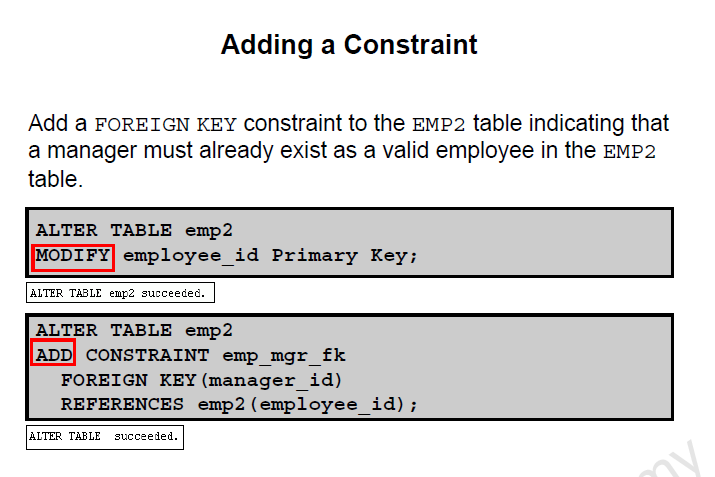
DROP constraints

-- Not modify THE STRUCTURE

ENABLE

DISABLE

ADD NOT NULL



ON DELETE CASCADE

Delete child rows when a parent key is deleted

ALTER TABLE EMPLOYEES

ADD CONSTRAINT em\_dt\_fk

FOREIGN KEY (department\_id)

REFERENCES departments (department\_id)

ON DELETE CASCADE

Allow parent key data that is referenced in the child table to be deleted but not updated

When parent row is deleted, all child rows that reference that parent key is also deleted.

-- i.e. if you delete the department ID in the department table all references in employees will be deleted

*What serious problem may occur?*

Different solutions when deleting parent data - #1

DROP TABLE DEPT\_100;

CREATE TABLE dept\_100

(emp\_id number (4) PRIMARY KEY,

Enter this to demonstrate

Lname varchar2(10),

Mgr\_id number (4) CONSTRAINT fk\_MGRd100

REFERENCES DEPT\_100

ON DELETE CASCADE

);

INSERT INTO DEPT\_100 VALUES (1,'ONE',NULL);

INSERT INTO DEPT\_100 VALUES (2,'two',1);

INSERT INTO DEPT\_100 VALUES (3,'333',1);

INSERT INTO DEPT\_100 VALUES (4,'444',2);

INSERT INTO DEPT\_100 VALUES (5,'555',2);

SELECT \* FROM DEPT\_100;

RESULT:

EMP\_ID LNAME MGR\_ID

---------- ---------- ----------

1 ONE

2 two 1

3 333 1

4 444 2

5 555 2

Remove manager 1

delete from dept\_100

where emp\_id = 1;

select \* from dept\_100;

**Here is the output…. WHAT HAPPENED?**

|  |  |
| --- | --- |
| 1 rows deleted. |  |
| no rows selected | 🡸 what happened. Why no rows left in the table |

Different solutions when deleting parent data -- #2

DROP TABLE DEPT\_100;

CREATE TABLE dept\_100

(emp\_id number (4) PRIMARY KEY,

Lname varchar2(10),

Mgr\_id number (4) CONSTRAINT fk\_MGRd100

REFERENCES DEPT\_100

ON DELETE SET NULL

);

INSERT INTO DEPT\_100 VALUES (1,'ONE',NULL);

INSERT INTO DEPT\_100 VALUES (2,'two',1);

INSERT INTO DEPT\_100 VALUES (3,'333',1);

INSERT INTO DEPT\_100 VALUES (4,'444',2);

INSERT INTO DEPT\_100 VALUES (5,'555',2);

SELECT \* FROM DEPT\_100;

RESULT:

EMP\_ID LNAME MGR\_ID

---------- ---------- ----------

1 ONE

2 two 1

3 333 1

4 444 2

5 555 2

Remove manager 1

delete from dept\_100

where emp\_id = 1;

select \* from dept\_100;

1 rows deleted.

EMP\_ID LNAME MGR\_ID

---------- ---------- ----------

Note the different effect

2 two

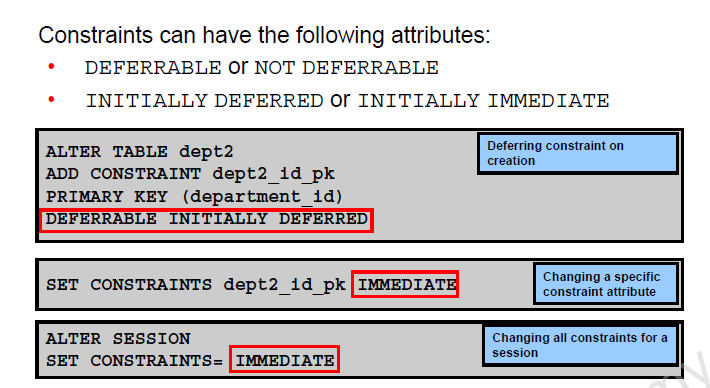
3 333

4 444 2

5 555 2

DEFERRING CONSTRAINTS

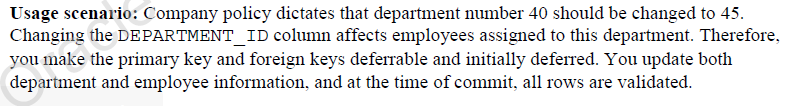
Used in transactions



This is a way of deferring the checking of constraints until the end of a transaction.

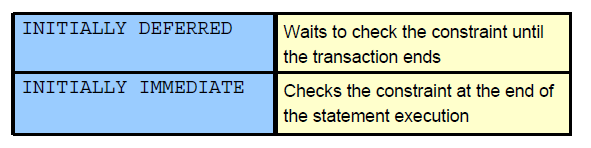
If at a commit the transaction is not correct and a constraint is violated the commit causes the transaction to roll back.

EXAMPLE of usage:



Differences

Initially Deferred – Initially Immediate



Create a table for testing and demoing

(next Page)

Drop table emp\_new\_sal;

CREATE TABLE EMP\_NEW\_SAL

(SALARY NUMBER

CONSTRAINT sal\_ck

CHECK (salary > 100)

**DEFERRABLE INITIALLY IMMEDIATE**, -- means will check immediately

BONUS NUMBER

CONSTRAINT bonus\_ck

CHECK (bonus > 0 )

**DEFERRABLE INITIALLY DEFERRED** -- deferred

);

**TESTING IT**:

ERROR: There is an immediate constraint on SALARY and 90 is a value too small

Do an insert.

INSERT INTO emp\_new\_sal

VALUES ( 90 , 5 );

The BONUS is not verified at this time as it is deferred.

**Test CONSTRAINT ON BONUS with -1**

INSERT INTO emp\_new\_sal

VALUES ( 100 , -1 ); 🡸 error still because salary not > 100

INSERT INTO emp\_new\_sal 🡸 insert ok even though bonus NOT > 0

VALUES ( 200 , -1 );

Waits for a COMMIT or until set the state back to IMMEDIATE

ALTER SESSION

Set constraints = immediate

TEST the action varying the INSERT

INSERT INTO emp\_new\_sal

VALUES ( 300 , -1 );

More Examples:

Example 2:

INSERT INTO emp\_new\_sal VALUES (100, -1);

Successfully insert the row.

COMMIT:

Causes an error as a constraint on bonus was violated. The transaction is rolled back.

Example 3:

SET CONSTRAINTS ALL DEFERRED;

Sets all constraints that can be deferred to that status.

Reissue insert

INSERT INTO emp\_new\_sal VALUES ( 90 , 5 );

Everything succeeds.

At COMMIT, both constraints are violated

Example 4:

Change status from deferred to immediate.

SET CONSTRAINTS ALL IMMEDIATE

NOTE:

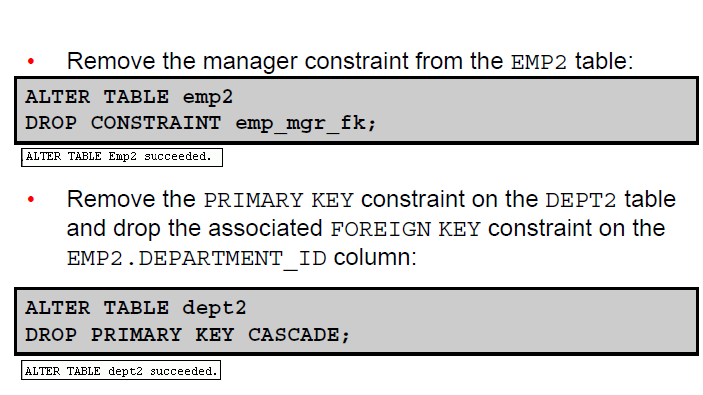
If you create a table with no deferred aspects, then the constraint is checked immediately.

You cannot set a constraint to DEFERRED that is not deferrable.

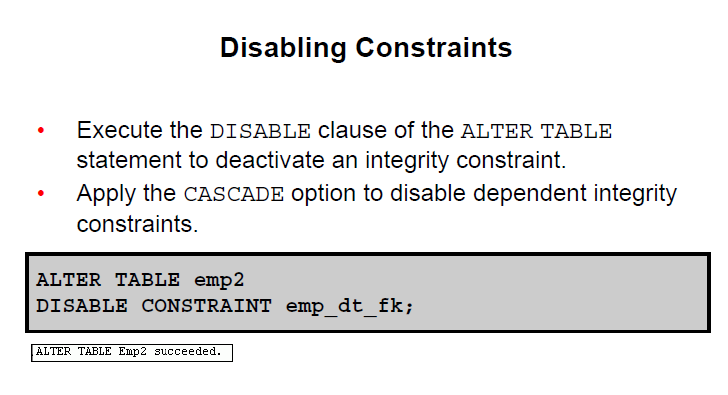
SET CONSTAINT newemp\_det\_pk DEFERRED

Dropping Constraint

2-18



Removes the foreign key references in the EMP table for department\_id which did reference dept2



OR

ALTER TABLE EMP2

DISABLE CONSTRAINT emp\_dt\_fk CASCADE

Disable is available on the CREATE table as well

Just for your information only

Create table divisions

(divno number constraint ck\_divno

CHECK (divno between 10 and 99)

DISABLE,

divname varchar2(9) constraint ck\_divname

CHECK (divname = UPPER(divname) -- ensures uppercase

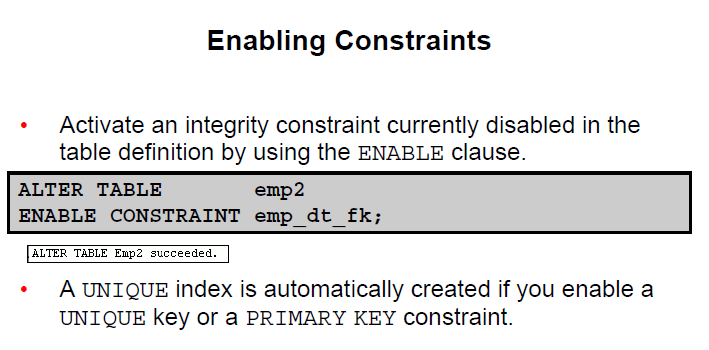
DISABLE,

Office varchar2(10) constraint ck\_office

CHECK (office in ('Toronto','New York','Boston','Montreal')

DISABLE

);



If you enable a constraint, it will apply to all the data in the table

If previously disable with a cascade it does NOT enable the foreign keys that are dependent on the PK. You may need to go back and fix that.