

Structured Query Language

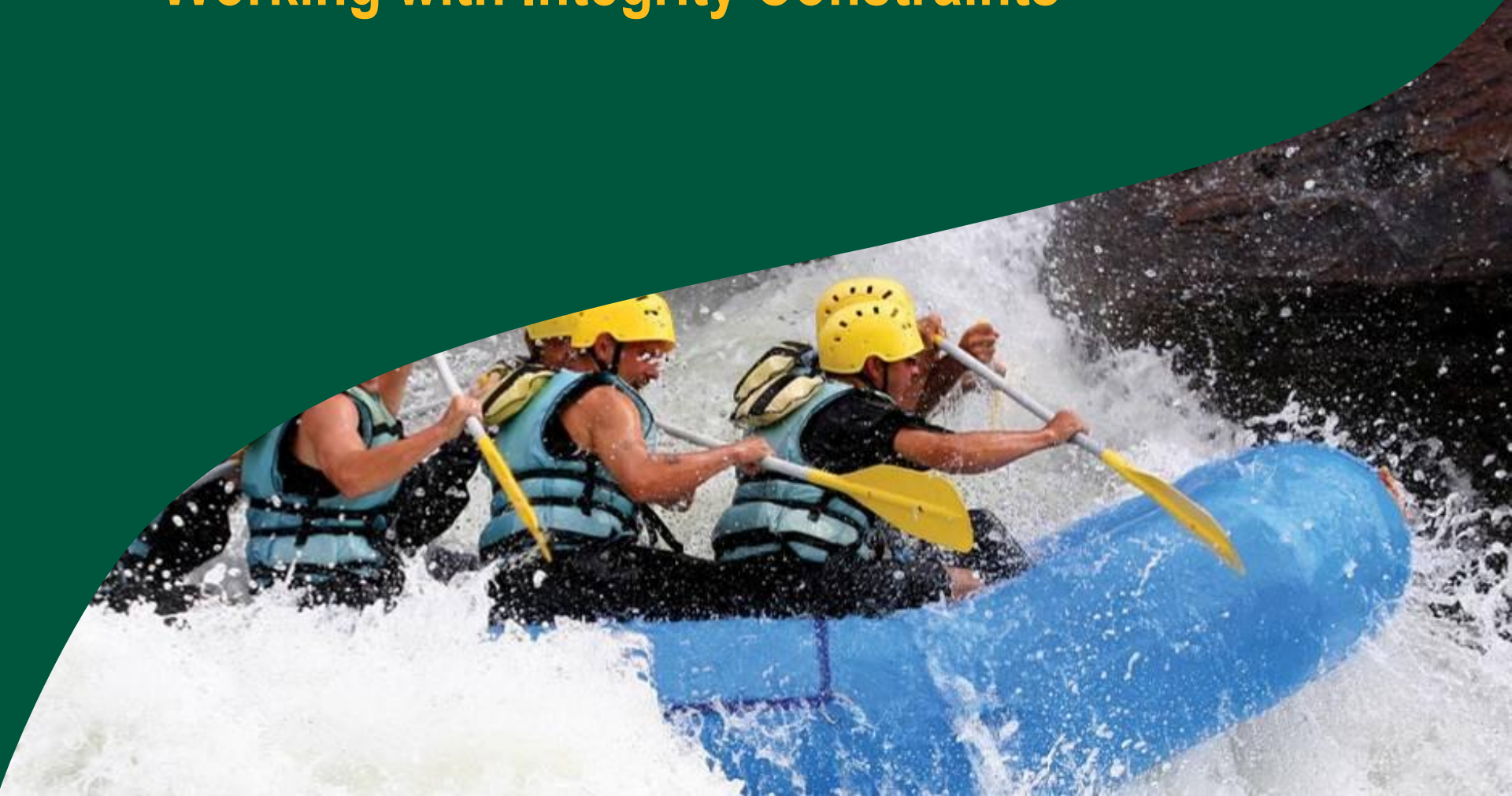
Session 2



Coverage

- Working with Integrity Constraints
- Altering Table Structure
- Creating table based on data of another table
- Inserting with Insert...Select
- Rename Table
- Removing Table
- Truncate Table
- Data Transaction Language

Working with Integrity Constraints



Introduction to Integrity Constraints

- It is important that in order to maintain data integrity data should adhere to some predefined business rules.
- Data integrity rules are used to enforce business rules associated with your database in order to prevent invalid data/information into the table
- Oracle implements data integrity rules through integrity constraints and database triggers

Types of Integrity Constraint

- » NOT NULL
- » Unique
- » Primary Key
- » Foreign Key
- » Check

NOT NULL Constraint

- A rule defined on a column to allow or disallow inserting or updating of a column value to NULL value
- By default all columns allow NULL values.
- A NOT NULL constraint requires that the column should contain a value

e.g. Employee name cannot contain Null value.
Every department has a name

Unique Constraint

- A rule defined on column(s) to allow for insert or update of a row only if it contains a unique value in that column(s)
- No two rows in the table can contain duplicate values for those column(s)
- Allows NULL values

e.g.

Telephone numbers are always distinct

Primary Key Constraint

- A rule defined to uniquely identify a row in the table
- No two rows can have a duplicate values on the primary key
- Does not allow Null values
- Each table can have at most one primary key

e.g

Employee No helps to identify employee related transactions uniquely

Referential Integrity / Foreign Key

- A rule defined on column(s) such that the value of that column(s) is the value of the primary key or unique value from the same or another table.
- The column on which the rule is defined is known as the foreign key
- Foreign key can have NULL values
- A composite foreign key can have all NULL, all non-null or partially null values

e.g.


The department number stored in Employee table should always refer the department number of Dept table

Terms used in Foreign key relationship

- **Referenced Key** : The primary key or unique key of the same table or another table
- **Foreign Key** : The column which references the Referenced key
- **Referenced or Parent table** : The table which is referenced by the foreign key
- **Dependent or Child table** : The table that defines the foreign key

Employee ~ Child / Dependent Table

Primary Key

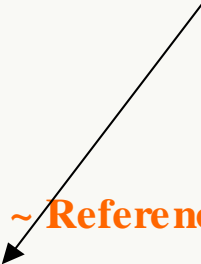


Emp No	Emp Name	Dept Code	Doj
1001	James	Acct	21-JUN-01
1002	Daniel	Mktg	14-JUN-04
1003	Mary	Acct	18-MAY-02
1004	Thomas	Acct	24-SEP-03
1005	Dick	Hrd	18-FEB-01

Foreign Key

Department ~ Referenced / Parent Table

**Primary Key
Referenced
Key**



Dept Code	Dept Name
Acct	Accounts
Mktg	Marketing
Hrd	Human Resource

Actions defined on Foreign Key

- **No Action :**

- The primary key cannot be updated or deleted if it is referenced by the foreign key. The data from the parent table cannot be deleted because data dependency exists on the child table.

- **Delete Cascade :**

- Will delete all the referenced rows in the child table , when primary key rows are deleted from the parent table.

- **Delete Set Null :**

- Will set the value of the columns to NULL when the primary key rows are deleted from the parent table.

Check Constraint

- Is a rule specified for performing specific simple validations
- It must be a Boolean expression
- A column can have multiple check options. Though it may not get evaluated in the same order of definition.

e.g.

Salary column cannot have a zero value

Default Column Values

- Default values can be specified for columns
- Default values are subject to all the integrity constraints defined on the column
- It is evaluated during parsing of the statement while INSERTING record in the table

e.g.

Mostly the employees are recruited on probation

Integrity Constraints definition level

- Integrity constraints can be defined at two levels :
 - Column
 - Table
- Column level constraint are defined along with column specification
- Table level constraints are defined after all the column specifications has been completed .
- If a constraint needs to refer multiple columns in the table then it should be defined at table level

Let's define the following constraint at column level

Create Table Empl

```
( empno number(3) primary key,  
  ename varchar2(15) not null,  
  contact_no number(10) unique,  
  deptno number(2),  
  emp_status char default 'P',  
  salary number(8,2) check(salary > 0),  
  join_date date default sysdate)
```

/

Defining constraints with user defined names

Create Table Empl

```
( empno number(3) constraint emp_pk PRIMARY KEY,  
  ename varchar(15) constraint name_blank NOT NULL,  
  contact_no number(10) constraint contact_unq UNIQUE,  
  deptno number(2),  
  emp_status char DEFAULT 'P',  
  salary number(8,2) constraint sal_zero CHECK(salary > 0),  
  join_date date DEFAULT SYSDATE  
)
```

Defining constraints at table level

```
Create Table Empl
(empno number(3),
ename varchar2(15)      NOT NULL,
contact_no number(10),
deptno number(2),
emp_status char         DEFAULT 'P',
salary number(8,2),
comm number(8,2),
join_date date         DEFAULT SYSDATE,
constraint emp_pk      PRIMARY KEY(empno),
constraint name_blank UNIQUE(ename),
constraint sal_zero   CHECK(salary > 0),
constraint sal_comm   CHECK(comm < salary)
)
```

Creating constraints on existing tables

**Alter table add (constraint <const_name>
<constraint>,..... ,.....)**

```
alter table dept  
add constraint dept_pk primary key(deptno);
```

- The constraint would be added only if it contains valid data
- More than one constraint can be added in the same command by separating with comma

Foreign key constraint

■ Column Level :

```
Create Table Empl  
(empno number(3),  
  ename varchar2(15) NOT NULL,  
  contact_no number(10),  
  deptno number(2) REFERENCES DEPT(DEPTNO),  
  emp_status char DEFAULT 'P',  
  salary number(8,2),  
  .....  
)
```

■ Table Level :

```
constraint emp_dept FOREIGN KEY(DEPTNO) references dept(no) or  
constraint emp_dept FOREIGN KEY(DEPTNO) references dept
```

Foreign key Rules

- The data-type and size of foreign key and the referenced primary key should be same
- Column name of foreign key and referenced primary key can be different
- Foreign key cannot be defined partially on the primary key

Delete Cascade and Delete Set Null

```
Create Table Empl
(empno number(3),
ename varchar2(15)          NOT NULL,
contact_no number(10),
deptno number(2)    REFERENCES DEPT(DEPTNO) on Delete Cascade,
emp_status char      DEFAULT 'P',
salary number(8,2),
.....
)
```

```
Create Table Empl
(empno number(3),
ename varchar2(15)          NOT NULL,
contact_no number(10),
deptno number(2)    REFERENCES DEPT(DEPTNO) on Delete Set Null,
emp_status char      DEFAULT 'P',
salary number(8,2),
.....
)
```


Disabling constraints

**Alter table <table_name>
disable (constraint <const_name>,.....);**

Alter table dept disable constraint dept_pk;

Alter table dept disable primary key;

- Primary key and Unique constraint cannot be disabled if it is being referenced by foreign keys.

Will report error :

ORA-02273: this unique/primary key is referenced by some foreign keys

Enabling constraints

**Alter table <table_name>
enable (constraint <const_name>,.....);**

Alter table dept enable constraint dept_pk;

Alter table dept enable primary key;

- The constraint would be enabled only if it contains valid data

Removing Constraints

Alter table <table_name>

Drop constraint <constraint name>

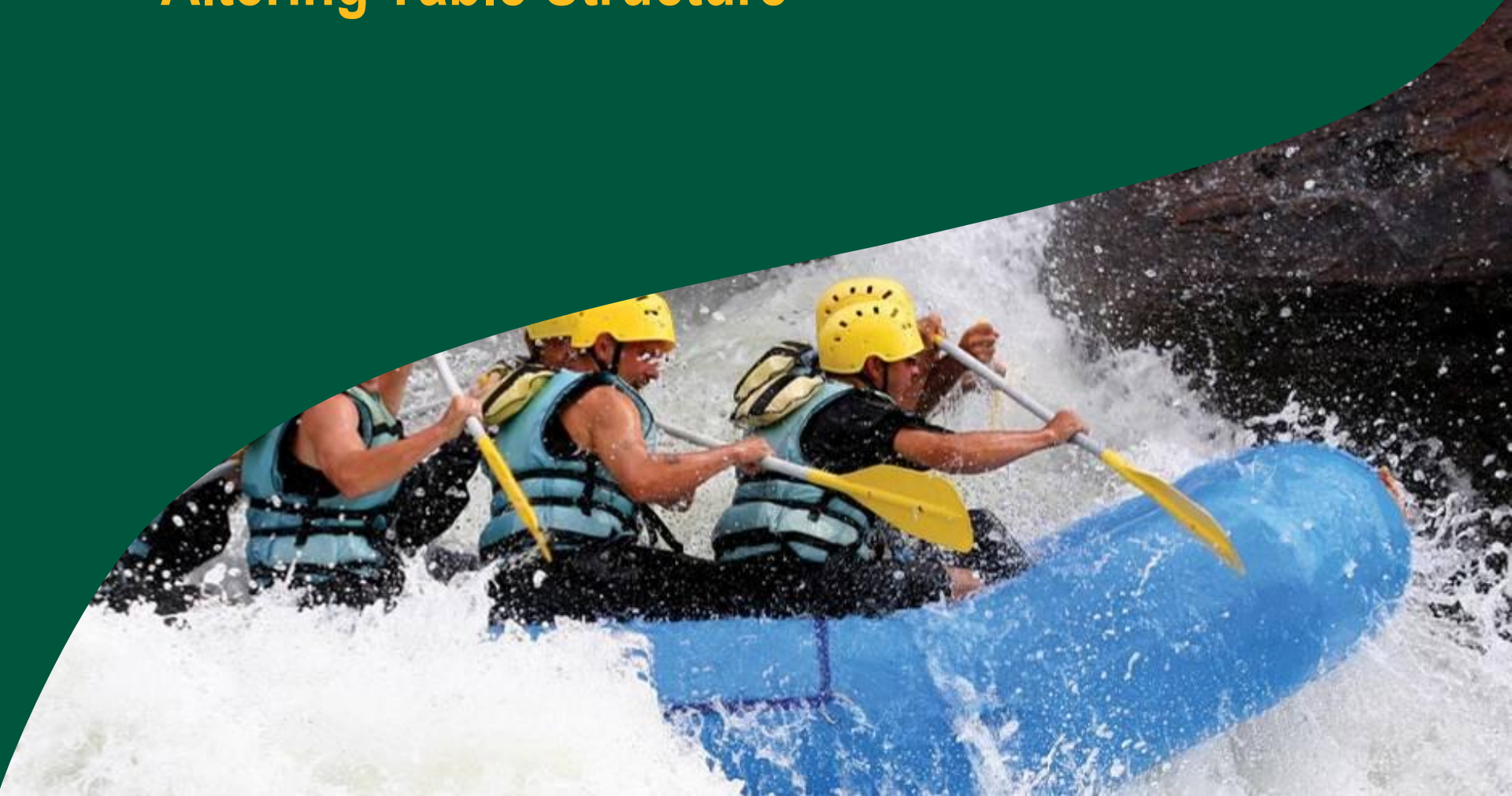
Alter table emp drop constraint emp_pk;

- Primary key and Unique constraint cannot be dropped if it is referenced by foreign keys. You need to drop the foreign keys constraints with the CASCADE option.

Alter table dept drop primary key cascade;

Alter table dept drop constraint dept_pk cascade;

Altering Table Structure



Adding a new column

```
Alter table <table_name>  
add (column_spec,...);
```

```
Alter table dept  
add (mgr varchar2(10));
```

Modifying existing column

```
Alter table <table_name>  
modify(column_specification1,  
column_specification2,...);
```

```
Alter table dept  
modify(mgr varchar2(15))
```

- Column to be modified must be empty :
 - to decrease column length less than occupied space or
 - for changing data type

Removing a Column

Alter table <table_name>

Drop column <column_name>;

Or

Alter table <table_name>

Drop (<column_list>;

```
alter table dept drop column mgr;
```

```
alter table dept drop ( mgr);
```

- A column cannot be dropped if it is used in defining integrity constraints.
- In order to drop such columns use CASCADE CONSTRAINTS option. It will drop all the constraints referencing the column and then drop the column

Creating table based on data of another table

**Create table <new_tablename>(column_name1,column_name2,,,,)
as
<select>**

```
create table dept_backup as  
select * from dept;
```

```
create table emp_job(employee_name ,job_title )  
as  
select  ename,job from emp  
where 1 = 2
```

- This will only create the table with the specified structure.

Inserting with Insert...Select

**Insert into <table_name>(<column_list>)
<select>**

```
insert into employee  
select * from emp;
```

- The structure of the target and source tables should match or do selective column(s) insert
- Columns with integrity constraint should be taken care of

```
insert into employee(empno,ename,deptno)  
select empno,ename,deptno from emp
```

Renaming table

Rename <old_table_name> to <new_table_name>;

Rename dept to department;

Removing Tables

Drop table <table_name>;

Drop table employee;

- Specify CASCADE CONSTRAINTS to drop all referential integrity constraints that refer to primary and unique keys in the dropped table.

Drop table dept cascade constraints;

TRUNCATE table

Truncate table <table_name>

- Deletes all the rows from the table
- Frees the storage space allocated to the table
- Cannot Rollback the changes

Transactions



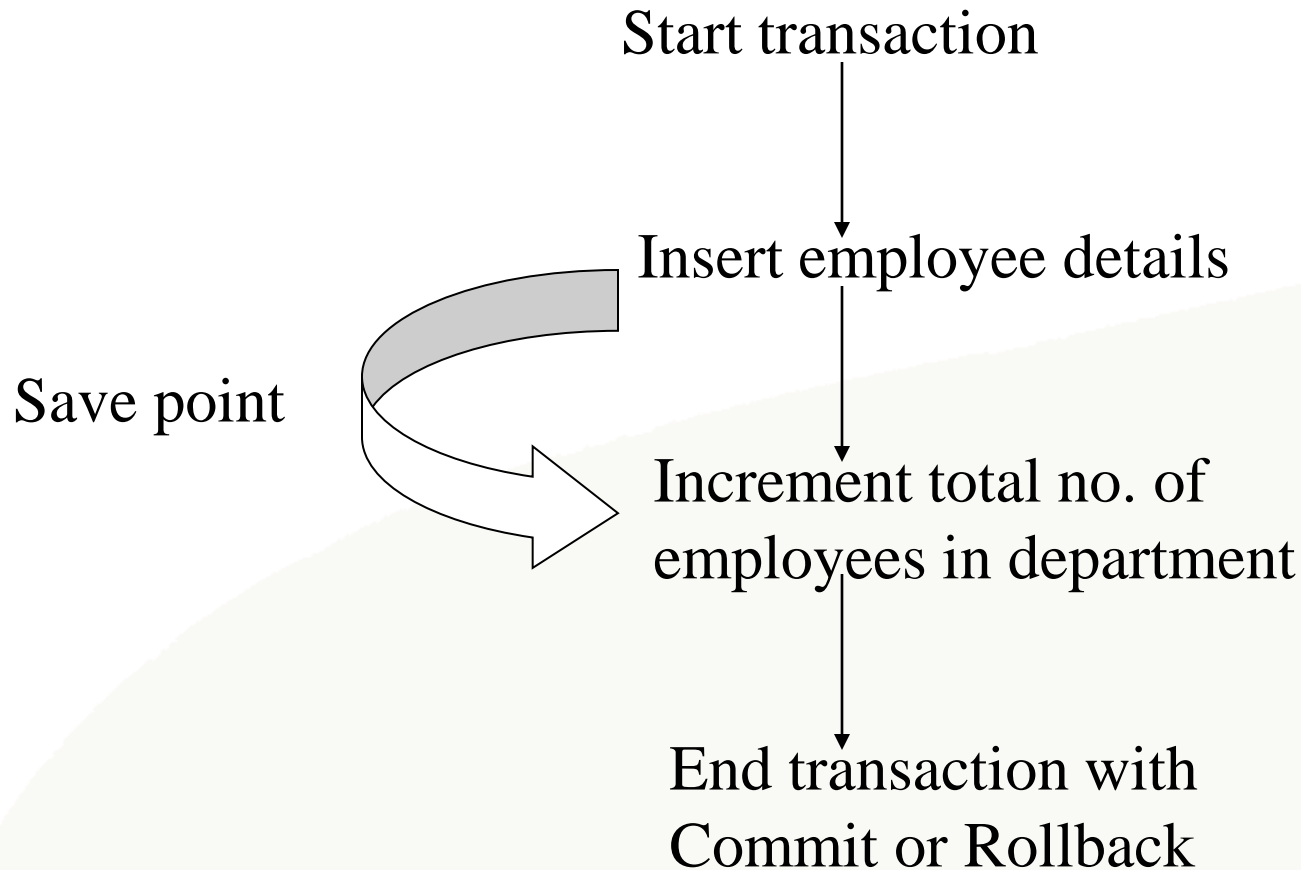
•Data Transaction Language

- Transaction : A set group of data manipulation commands is known as transaction.

E.g. When a employee joins, a transaction would be completed when his record is entered in employee table and total number of employees in department table is incremented by 1 for the respective department

- Commit - Makes the DML changes of the transaction permanent
- Rollback - undo the changes made through DML
- Savepoint - Intermediate points for rollback. A long transaction can be broken at different levels to ease COMMIT or ROLLBACK

Transaction - Employee Joins



Transaction

- Begin when the first DML SQL statement is executed
- End with one of the following events:
 - A COMMIT or ROLLBACK statement is issued
 - A DDL or DCL statement executes (automatic commit)
 - The user exits *iSQL*Plus*
 - The system crashes

Using Savepoint

Insert

Insert

Savepoint this_is_insert

Update....

Update....

Savepoint this_is_update

Delete

Rollback to this_is_insert

Implicit transaction processing

- An automatic commit occurs under the following circumstances:
 - DDL statement is issued
 - DCL statement is issued
 - Normal exit from SQL*Plus, without explicitly issuing COMMIT or ROLLBACK statements
- An automatic rollback occurs under an abnormal termination of SQL*Plus or a system failure.

State of Data Before Commit

- The previous state of the data can be recovered by using ROLLBACK.
- The current user can review the updated data
- Other users cannot view the updated data
- The affected rows are locked

State of Data after COMMIT

- Data changes are made permanent in the database.
- The previous state of the data is permanently lost.
- All users can view the updated data
- All savepoints are erased.
- Locks on the affected rows are released

State of Data after ROLLBACK

- Data changes are undone.
- Previous state of the data is restored.
- Locks on the affected rows are released.

Set transaction...

- Set transaction Read Only is a transaction level read consistence
- Set Transaction Read Write is a statement level read write

Set transaction Read Only

Select .. Count(*) ---- 20

Record inserted and
committed by another
user session

Select Sum(sal),Avg(sal)...

The result would be for 20
employees

Commit;

21 st record will be visible

Thank You !!!

