Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Structured Query Language

Coverage

- Introduction to SQL
- Categories of SQL commands
- Introduction to SQL * PLUS
- Creating and Managing Tables
- Retrieving Data
- Manipulating Data
- Comments
- User Interactions
- Working with Integrity Constraints
- Altering Table Structure
- Data Transaction Language

Coverage

- Retrieving Data
- Performing Arithmetic Calculations
- Handling NULL Values
- Concatenating Colums
- Giving Alias names to Columns
- Relational Operators
- Logical Operators
- Ordering Data
- Displaying Data from Multiple Table (JOINS)
- Set Operators

Coverage

- Types of SQL Functions
  - Scalar
  - Group / Aggregate / Columnar
- String functions
- Date functions
- Numeric functions
- Common functions
- Conversion functions
- Number and Date Formats

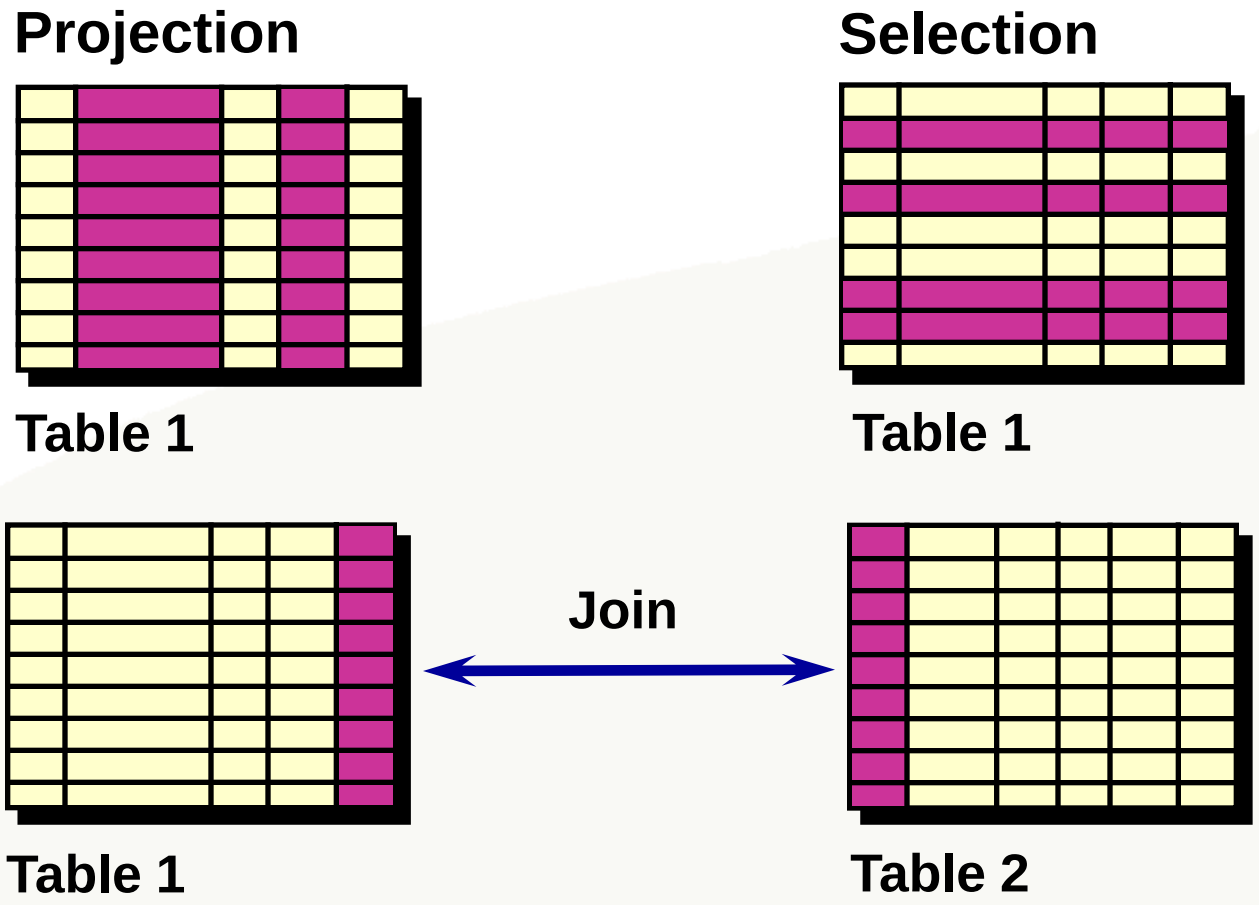
Coverage

- Grouping Data
- Sub Queries
- Nested Subquery
- Inline view
- Correlated query
- Working with Views
- Working with Indexes
- Working with Sequences
- Using Synonyms

Introduction to SQL

- Structured Query Language is a database sub-language.
- It is used to interact with database in order to store, manipulate or retrieve data
- It is also pronounced as SQL or Sequel
- It is a standard language for RDBMS
- It is a non-procedural language
- It is based on the relational model proposed by Dr. E.F. Codd

Capabilities of SQL `SELECT` Statements



Categories of SQL commands

Data Definition Language (DDL)	Create, Alter, Drop
Data Manipulation Language (DML)	Insert , Update, Delete
Data query Language	Select
Data Control Statements	Grant, Revoke
Transaction Control Statements	Commit, Rollback, Savepoint, Set transaction
Session Control Statements	Alter session
System Control Statements	Alter system
Embedded SQL Statements	

Introduction to SQL*PLUS

- It is a product of Oracle
- Works in conjunction with SQL database language and its procedural language PL/SQL
- Allows you to enter,edit, store and retrieve data
- Perform calculations,format, store, and print query results in the form of reports
- Access and copy data between SQL databases
- Prompt for user interaction
- Perform database administration

Getting Started with SQL * PLUS

- **Invoking SQL*PLUS**
  - Command Line Interface
    - SQLPLUS [username/password][@connectstring]
  - Graphical Interface
    - Start >Programs->Oracle->Application Development->SQL Plus
- **Leaving SQL*PLUS**
  - Type Exit on SQL prompt

    or
  - Close the window

Connecting to Oracle through SQL *PLUS

Getting Started….

- Cursor will appear after the SQL prompt for accepting the command
- Words in the command are separated by space or tab
- The manner in which you continue a command on a additional line, end a command or execute a command will depend on the kind of SQL command given .

Kind of commands given at SQL prompt

SQL PLUS or (SQL)	A data sub-language for working with data in the database
PL/SQL blocks	A procedural language for working with data in the database
SQL * PLUS	Incorporates commands related to formatting query results , setting options, editor commands

SQL Statements Vs SQL*Plus commands

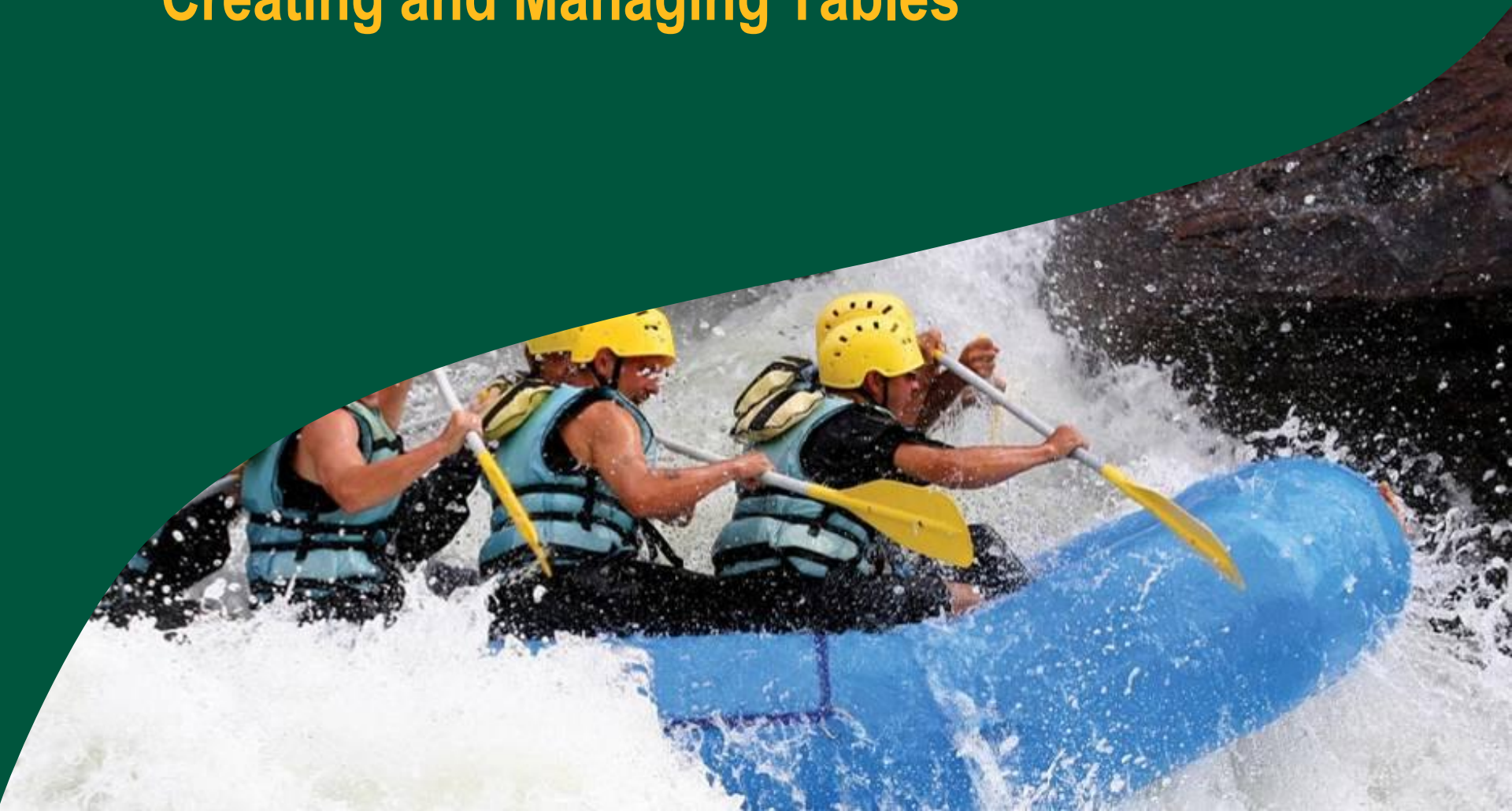
- SQL
  - It is a language to communicate with database
  - Follows ANSI standard
  - Keywords cannot be abbreviated
  - Manipulates data and table definitions in the database
  - Does not have/require continuation character to continue command on next line
  - Uses functions to perform formatting
- SQL *Plus
  - It is an environment. Sends SQL commands to server for processing
  - It is Oracle's proprietary product
  - Keywords can be abbreviated
  - Data manipulation not allowed
  - Runs on Browser
  - May be Centrally loaded
  - Dash (-) is used as continuation character
  - Uses commands to perform formatting

Command termination Character

Command	Termination Character
SQL PLUS	; /
PL / SQL	/
SQL * PLUS	Enter

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Creating and Managing Tables



Creating a Simple Table

```
Create Table <table_name>
( column_name1 data_type(size),
  column_name2 data_type,
  column_name3 data_type(size),
   ……..
 column_name data_type(size)
 );
```

Built-In Datatypes

VARCHAR2(size)	Variable-length character string having maximum 4000 bytes.
CHAR(size)	Fixed-length character data with maximum 2000 bytes
NUMBER(p,s)	Number having precision p and scale s. Precision can range from 1 to 38. Scale can range from –84 to 127

Built-In Datatypes

DATE	Date in the default format as DD-MON-RR. The date is internally stored as CENTURY, YEAR, MONTH, DAY, HOUR, MINUTE and SECOND
RAW(size)	Raw binary data of maximum 2000 bytes
LONG	Character data of variable length upto 2 gigabytes
LONG RAW	Raw binary data of variable length upto 2 gigabytes
BFILE	Contains locator to a binary file (maximum 4 gigabytes) stored outside the database

Built-In Datatypes

BLOB	A binary large object of maximum 4 gigabytes
CLOB	A character large object containing single-bytes character of maximum 4 gigabytes

Naming conventions

- Names of object or column names cannot be greater than 30 bytes
- Special symbols like ; / *? not to be used
- Names are not case sensitive
- Must begin with alphabetic character unless surrounded by double quotation marks.
- Cannot be an Oracle reserved word
- Give meaningful names

Create Table

```
Create Table Employee
( empno number(3),
  ename varchar2(15),
  deptno number(2),
  grade char,
  salary number(8,2),
  join_date date);
```

Viewing Structure of Table

Desc <table_name>

e.g

Desc employee

Inserting Data

Insert into <table_name>
(column1, column2, column3,…)
values (column1_value,column2_value,…..)

e.g.

```
Insert into Employee
values(101,'John',10,'M',9000,'01-JAN-01');
```

Insert - Character strings and date values

- Character strings and date values are enclosed in single quotation marks.
- Character values are case sensitive
- Date values are format sensitive.
- The default date format is DD-MON-RR

Inserting Data – NULL values

Insert into <table_name>

values (column1_value, column2_value, NULL, column3_value,NULL, column4_value …..)

e.g.

```
Insert into Employee
values(101,'John',NULL,'M',9000,NULL);
```

Inserting data through User Interaction

Insert into <table_name>
values (&var_name,&var_name,…..)

e.g.

```
Insert into Employee
values(&emp_no,'&emp_name',&dept_no,'&grade',&salary,
'&doj')
/
```

Inserting data partially

Insert into
<table_name>(column_name1,column_name2)
values (column_value1,column_value2);

e.g.

```
Insert into Employee(empno,ename)
values(105,'John R');

Insert into Employee(empno,ename)
Values(&emp_no,'&emp_name');
```

Retrieving data

- Data is retrieved by using queries.
- Query is an operation that retrieves data from one or more tables or views.

Select */<column_list>
From <table_name>
Where <conditions>
Group By <column_list>
Having <condition>
Order By <column_list>

Simple Examples of Select statement

```
Select * from dept;

Select * from emp
Where deptno = 10;

Select empno,ename, salary
from emp
Where empno = 7900;
```

Editing data

To make changes to existing rows

Update <table_name>
Set col_name_1 = <new_value>,
col_name_2 = <new_value>
Where <condition>

```
update emp
set deptno = 10
where empno = 7900;

update emp
set deptno = 10,
    mgr = 7839
where empno = 7900;
```

Deleting records

Removing existing rows

Delete from <table_name>
Where <condition>;

Delete from dept
where deptno = 40;

Removing a Table

Drop table <table_name>;

Drop table dept;

Putting Comments

Remark	Single line comments can be placed after the keyword Remark at the beginning of line
/* */	Mutli-line comments can be placed between /*…. and */
--	Single line comments can be placed after – anywhere in the line. The text placed after – is treated as comments till the end of line

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Thank You !!!