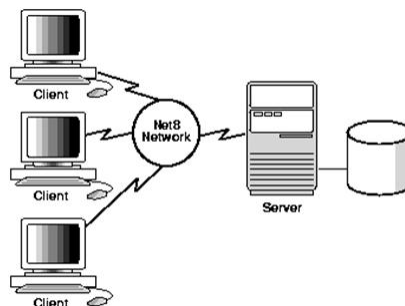


# Chapter 2

## SQL Basics

### The SQL\*Plus Environment

- Client/server environment
  - front end
    - presentation of data to user
  - back end
    - the data store
- SQL\*Plus
  - client-side software
  - sends SQL commands to and receives results from Oracle
- Oracle
  - DBMS that runs on a server
- OracleNet
  - establishes network sessions and transfers data





## Schemas

---

- A **schema** is a collection of objects owned by a user account
  - when you connect, you login to your schema
- SHOW USER
  - SQL\*Plus command to verify who the current user is

```
SQL> show user  
USER is "SCOTT"
```

3



## DESCRIBE Command

---

- A SQL\*Plus command that shows a table's structure
- Lists each field's name, nullability, datatype
- Can be abbreviated as DESC

4



## Writing SQL Statements

- Can be on one or more lines
- Are not case sensitive
- Keywords cannot be abbreviated or split across lines
- Enhance readability by
  - placing each clause on a separate line
  - using indentation
  - are terminated by ;

5



## Common Data Types

Datatype	Description
VARCHAR2( <i>size</i> )	Variable-length character data; size required; size must not exceed 4000.
CHAR( <i>size</i> )	Fixed-length character data; size defaults to 1 and cannot exceed 2000
NUMBER( <i>prec, scale</i> )	Variable-length numeric data; fixed and floating point; <b>precision</b> is total size, <b>scale</b> is digits to right of decimal point
DATE	Date/time values; range from Jan 1, 4712 BC to Dec 31, 4712 AD

6



## Other Data Types

Datatype	Description
TIMESTAMP	Date/time values with higher precision and timezone information
ROWID	Hexadecimal value representing the unique address of a row in a table
LONG	Variable-length character data up to 2 gigabytes
CLOB	Single-byte character data up to 4 gigabytes
RAW, LONG RAW, BLOB, BFILE	Binary datatypes

7



## The SELECT Statement

```
SELECT [DISTINCT] {*, column [alias],...}
FROM table;
```

- Used to retrieve columns from one or more tables
- SELECT identifies which columns
- FROM identifies which table to obtain data from
- Statement syntax (nib)
  - [brackets]
  - {braces}
  - ellipsis ...

8

# Capabilities of SQL SELECT Statements



## Selection


Table 1

## Projection


Table 1

## Join


Table 1




Table 2

9

# Practice Time



- Show the structure of the Article table
- Show all fields for each article
- Show each article's title, date published and number of words

10



## Eliminating Duplicate Rows

```
SELECT [DISTINCT] {*, column [alias],...}
FROM   table;
```

- By default queries return all rows, even those with duplicate information
- Use **DISTINCT** to suppress duplicate **rows**

```
SQL> SELECT writerid FROM article ORDER BY writerid;
```

```
SQL> SELECT DISTINCT writerid FROM article ORDER BY writerid;
```

```
SQL> SELECT DISTINCT writerid, title FROM article ORDER BY writerid;
```

- Show an unduplicated list of types of articles that exist in the article table

11



## Editing a SQL Statement

- SQL buffer
  - contains the most recently typed SQL statement
  - can re-execute it with **/**
  - **\*** denotes current line
- Edit command
  - launches the default text editor

```
SQL> edit
SQL> edit first.sql
```

12



# Scripts

- Script file
  - an unformatted text file containing one or more SQL statements and/or SQL\*Plus commands
  - defaults to .sql extension
  - default script location `c:\oracle\product\10.2.0\client_1\BIN`
- Start command
  - runs a specified script

```
SQL> start first.sql  
SQL> @first.sql
```

13



# Script Files

- Save command
  - saves the current SQL statement as a text file/script

```
SQL> SAVE filename
```

- Get command
  - opens the specified script file & places in SQL buffer

```
SQL> GET filename
```

14



## The WHERE Clause

```
SELECT      [DISTINCT] {*| column [alias], ...}  
FROM        table  
[WHERE      condition(s)];
```

- A condition that restricts which **rows** participate in a SQL statement
- Conditions are built with a comparison operator and a value
- A condition must compare two values of the same datatype

15



## Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>, !	Not equal to

16



## WHERE Clause Practice

- Show the title, issue and length of articles shorter than 1500 words.
- Show the title, issue and length of the business articles.
- Show the last name, first name and phone number for the freelancer writers.

17



## More Comparison Operators

Operator	Meaning
<b>BETWEEN ...AND...</b>	<b>Between two values (inclusive)</b>
<b>IN(list)</b>	<b>Match any of a list of values</b>
<b>LIKE</b>	<b>Match a character pattern</b>
<b>IS NULL</b>	<b>Is a null value</b>

18



## The BETWEEN Comparison Operator

- An alternative way to select rows based on a **range** of values

```
SQL> SELECT title, length
2 FROM article
3 WHERE length BETWEEN 500 AND 1500;
```

- Equivalent to

```
SQL> SELECT title, length
2 FROM article
3 WHERE _____;
```

19



## The IN Comparison Operator

- Selects rows based on items in a **predefined list**

```
SQL> SELECT *
2 FROM article
3 WHERE type IN ('BUS', 'LAW');
```

- Equivalent to

```
SQL> SELECT *
2 FROM article
3 WHERE _____;
```

20



## The LIKE Operator

- Selects rows based on a **pattern** you specify
  - pattern provides flexibility when exact value is not known
- Use the **LIKE** keyword and wildcard symbol(s)
  - %

```
SQL> SELECT title, type
2 FROM article
3 WHERE title LIKE '%$%';
```

- Show name and phone of all writers.
- Show name and phone of writers in area code 710.

21



## Evaluating Null Values

- NULL represents a missing or unknown value
- IS NULL operator
  - used to test for null values

```
SQL> SELECT ln, fn, phone
2 FROM writer
3 WHERE phone IS NULL;
```

- don't try compare a value to NULL

```
SQL> SELECT ln, fn, phone
2 FROM writer
3 WHERE phone = NULL;
```

- Which writers don't have a contact?
- Which articles don't have a writerid?

22



## Logical Operators

- Used when constructing a query containing multiple criteria

Operator	Meaning
NOT	Returns TRUE if the following condition is FALSE
AND	Returns TRUE if <i>both</i> conditions are TRUE
OR	Returns TRUE if <i>either</i> condition is TRUE

- Shown in order of **precedence**
  - may need parentheses to override precedence
  - $3 + 4 * 2$

23



## Logical Operators: Examples

```
SQL> SELECT title, type, length
2 FROM article
3 WHERE type = 'BUS' AND length <1500;
```

```
SQL> SELECT title, type, length
2 FROM article
3 WHERE type = 'BUS' OR length <1500;
```

24



## Logical Operators: Precedence

- Which are either political or business articles that are shorter than 1500 words?

```
SQL>SELECT title, type, length
2 FROM article
3 WHERE type='POL' OR type='BUS' AND length<1500;
```

9 rows

```
SQL>SELECT title, type, length
2 FROM article
3 WHERE (type='POL' OR type='BUS') AND length<1500;
```

7 rows

25



## Logic Tables

AND	TRUE	FALSE	NULL	OR	TRUE	FALSE	NULL	NOT	
TRUE	TRUE	FALSE	NULL	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	NULL	FALSE	TRUE
NULL	NULL	FALSE	NULL	NULL	TRUE	NULL	NULL	NULL	NULL

26



## Column Alias

- By default, a field's name is used as column heading
- Can specify a different column heading
- Multi-word aliases must be enclosed in double-quotes

```
SQL> SELECT phone AS Phone# FROM writer;

SQL> SELECT phone Phone# FROM writer;

SQL> SELECT phone "Phone Number"
FROM writer;
```

27



## The ORDER BY Clause

```
SELECT      [DISTINCT] {*| column [alias], ...}
FROM        table
[WHERE      condition(s)]
[ORDER BY  column [{ASC|DESC}] ], ...];
```

- Used to **sort** the returned rows
  - can sort ascending or descending
  - can sort by multiple columns

```
SQL>SELECT ln, fn, freelancer
2 FROM writer
3 ORDER BY freelancer;
```

```
SQL>SELECT ln, fn, freelancer
2 FROM writer
3 ORDER BY freelancer, ln, fn;
```

- can cite column alias
- can cite column position

28



## ORDER BY and NULLS

- Can use **NULLS FIRST** or **NULLS LAST** clause to control placement of nulls

```
SQL> SELECT ln, phone
  2 FROM writer
  3 ORDER BY phone;
LN          PHONE
-----
Cox          (210) 783-5415
Johnson     (210) 895-2046
Martinez    (544) 332-7788
Seeger      (576) 423-0932
...
Cohen       (910) 338-1875
Waldeck     (917) 361-8181
Kim         (917) 729-5364
Nilsson
Lawton
```

```
SQL> SELECT ln, phone
  2 FROM writer
  3 ORDER BY phone NULLS FIRST;
LN          PHONE
-----
Nilsson
Lawton
Cox          (210) 783-5415
Johnson     (210) 895-2046
Martinez    (544) 332-7788
Seeger      (576) 423-0932
...
Cohen       (910) 338-1875
Waldeck     (917) 361-8181
Kim         (917) 729-5364
```

29



## ORDER BY Clause Practice

- Show title, type, issue, length of each article sorted by length with longest article on top
- Show title, type, issue, length of each article sorted by type and secondarily sorted by date with most recent on top
  - note the chronological sort...

30