

Using SQL Queries to Insert, Update, Delete, and View Data

Date Retrieval from a single table & Calculations

Wednesday 2/4/2015

© Abdou Illia

MIS 4200 - Spring 2015

Database Object Privileges

Object Type(s)	Privilege	Description
Table, Sequence	ALTER	Allows the user to change the object's structure using the ALTER command
Table, Sequence	DROP	Allows the user to drop (delete) the object
Table, Sequence	SELECT	Allows the user to view table records or select sequence values
Table	INSERT, UPDATE, DELETE	Allows the user to insert, update, or delete table records
Any object	ALL	Allows the user to perform all possible operations on the object

Table 3-5 Commonly used Oracle10g object privileges

SQL GRANT command

```
GRANT privilege1, privilege2, ...  
ON object_name  
TO user1, user2, ...
```

SQL REVOKE command

```
REVOKE privilege1, privilege2, ...  
ON object_name  
TO user1, user2, ...
```

Lesson B Objectives Chapter 3B

After completing this lesson, you should be able to:

- Write SQL queries to retrieve data from a single database table
- Create SQL queries that perform calculations on retrieved data
- Use SQL group functions to summarize retrieved data

Retrieving Data From a Single Database Table

- Syntax
`SELECT columnname1, columnname2, ...
FROM ownertype.tablename
[WHERE search_condition];`
- Retrieve all of columns
 - Use asterisk (*) as wildcard character in SELECT clause
 - `SELECT *`
 - `FROM tablename`

Suppressing Duplicate Rows

- SQL DISTINCT qualifier
 - Examines query output before it appears on screen
 - Suppresses duplicate values
- Syntax
 - `SELECT DISTINCT columnname;`

Using Search Conditions in SELECT Queries

- Use search conditions to retrieve rows matching specific criteria
 - Exact search conditions
 - Use equality operator
 - Inexact search conditions
 - Use inequality operators
- Search for NULL or NOT NULL values
 - `WHERE columnname IS NULL`
 - `WHERE columnname IS NOT NULL`

Using Search Conditions in SELECT Queries (continued)

- IN comparison operator
 - Match data values that are members of a set of search values
- LIKE operator
 - Use to match part of character string
 - Syntax
 - WHERE *columnname* LIKE '*string*'
 - Character string may contain wildcard character %, or _, or both

Sorting Query Output

- ORDER BY clause
 - Sort query output
 - Syntax for select with ordered results

```
SELECT columnname1, columnname2, ...
FROM ownername.tablename
WHERE search_condition
ORDER BY sort_key_column;
```
 - Sort can be ascending or descending
 - Can specify multiple sort keys

Using Calculations in SQL Queries

- Perform many calculations directly within SQL queries
 - Very efficient way to perform calculations
- Create SQL queries
 - Perform basic arithmetic calculations
 - Use variety of built-in functions

Performing Arithmetic Calculations

- Perform arithmetic calculations on columns that have data types
 - NUMBER
 - DATE
 - INTERVAL
- SYSDATE pseudocolumn
 - Retrieves current system date
- Use + and – to calculate differences between dates

Oracle 10g SQL Functions

- Built-in functions perform calculations and manipulate retrieved data values
- Called single-row functions
 - Return single result for each row of data retrieved
- To use:
 - List function name in SELECT clause followed by required parameter in parentheses

Oracle 10g SQL Group Functions

- Group function
 - Performs operation on group of queried rows
 - Returns single result such as column sum
- To use:
 - List function name followed by column name in parentheses

Using the COUNT Group Function

- COUNT group function
 - Returns integer representing number of rows that query returns
- COUNT(*) version
 - Calculates total number of rows in table that satisfy given search condition
 - Includes NULL values.
- The COUNT(*columnname*) version
 - Does not include NULL values

Using the GROUP BY Clause to Group Data

- GROUP BY clause
 - Group output by column with duplicate values
 - Apply group functions to grouped data
- Syntax
 - GROUP BY *group_columnname*;
 - Follows FROM clause
- All columns listed in SELECT clause must be included in GROUP BY clause

Using the HAVING Clause to Filter Grouped Data

- HAVING clause
 - Place search condition on results of queries that display group function calculations
- Syntax
 - HAVING *group_function* *comparison_operator* value
- Example
 - HAVING sum(capacity) >= 100

Creating Alternate Column Headings

- Column headings for retrieved columns are names of database table columns
- Specify alternate output heading text

```
SELECT columnname1 "heading1_text ",  
columnname2 "heading2_text", ...
```

Creating Alternate Column Headings (continued)

- Alias
 - Alternate name for query column
 - Syntax
 - ```
SELECT columnname1 AS alias_name1...
```

---

---

---

---

---

---

---

---

### Modifying the SQL\*Plus Display Environment

- SQL\*Plus page consists of:
  - Specific number of characters per line
  - Specific number of lines per page
- linesize property
  - Specifies how many characters appear on line
- pagesize property
  - Specifies how many lines appear on page
- Modify using environment dialog box

---

---

---

---

---

---

---

---

## Formatting Data Using Format Models

- TO\_CHAR function
  - Convert column to character string
  - Apply desired format model to value
  - Syntax
    - `TO_CHAR(column_name, 'format_model')`
  - Use for data types
    - DATE
    - INTERVAL
    - NUMBER

---

---

---

---

---

---

---

---