IT2201 / IT2601 / IT2564 / IT2621 / IT2521 / IT2323

Database Management Systems

Unit 8 (Part A)

Structured Query Language (Basic SELECT)

1

Unit Objectives

- □ At the end of this unit, you should be able to
 - Use SQL to query and update databases.
 - Use SQL to define and create databases.

2

Unit 8a : SQL (Basic SELECT)

Page 1

AY2018/2019 S1

Topics

- In Unit 8a:
 - Overview of SQL
 - DML (Querying the database)
 - Basic SELECT statement
- □ In Unit 8b:
 - DML (Querying the database)
 - Advanced SELECT statement
- □ In Unit 8c:
 - DML (Updating the Database)
 - INSERT statement
 - UPDATE statement
 - DELETE statement
 - DDL (Defining the Database)
 - CREATE, ALTER statement

2

Overview of SQL

- Background
 - A particular language that has emerged from the development of the relational model is SQL.
 - SQL has become the standard relational database language.
 - In 1986, a standard for SQL was defined by ANSI, which was subsequently adopted in 1987 as an international standard by the ISO.
- What is SQL
 - It is a comprehensive Database Language
 - It has 2 major components:
 - Data Manipulation Language (DML)
 - Data Definition Language (DDL)

4

Unit 8a : SQL (Basic SELECT)

Page 2
AY2018/2019 S1

Overview of SQL

- It is a <u>non-procedural language</u>.
- SQL does not contain flow control commands.
 - It can be issued interactively or embedded within an application program.
- It can be used by a range of users
- An ISO standard now exists for SQL, making it both the <u>formal</u> and <u>de facto</u> standard language for relational databases

5

Writing SQL Commands

- SQL statement consists of reserved words and userdefined words.
 - Reserved words are a fixed part of SQL and must be spelt exactly as required and cannot be split across lines.
 - User-defined words are made up by user and represent names of various database objects such as relations, columns, views.

6

Unit 8a : SQL (Basic SELECT)

Page 3

AY2018/2019 S1

Writing SQL Commands

- Most components of an SQL statement are case insensitive, except for literal character data.
- More readable with indentation and lineation:
 - Each clause should begin on a new line.
 - Start of a clause should line up with start of other clauses.
 - If clause has several parts, should each appear on a separate line and be indented under start of clause.

7

SELECT Statement

- The SELECT Statement
 - SELECT is the most important and the most complex SQL statement.
 - It can be used
 - to retrieve and display data from one or more tables.
 - as part of an **INSERT** statement to produce new rows.
 - as part of UPDATE / DELETE statement to update/delete data.

8

Unit 8a : SQL (Basic SELECT)

Page 4
AY2018/2019 S1

SELECT statement

Syntax :

SELECT [DISTINCT] column_list

FROM table_name

{[INNER JOIN table_name ON condition]}

[WHERE condition]
[GROUP BY column_list]
[HAVING condition]

[ORDER BY column_list [DESC]]

Only SELECT & FROM are mandatory

Order of the clauses cannot be changed

۵

SELECT statement

SELECT Specifies which columns are to

appear in output

FROM Specifies table to be used

{[INNER JOIN .. ON ..]}

Specifies other table(s) to be joined. Repeats

for each additional table.

[WHERE] Filters rows

[GROUP BY] Forms groups of rows with same

column value

[HAVING] Filters groups subject to some

condition

[ORDER BY] Specifies the order of the rows in

the output

10

Unit 8a : SQL (Basic SELECT) Page 5 AY2018/2019 S1

SELECT ... FROM clause

[DISTINCT] column_list **SELECT**

FROM table_list

Retrieve full details of all customers

- Use * to denote <u>ALL</u> columns OR specify each column explicitly
 - select * from customer;
 - □ Select customer_num, fname, Iname, address, zipcode from customer;

customer_num	fname	Iname	address	zipcode
101	Ludwig	Pauli	213 Erstwild Court	94086
102	Carole	Sadler	785 Geary St	94117
103	Philip	Currie	654 Poplar	94303

3 rows selected

11

SELECT ... FROM clause

[DISTINCT] column_list **SELECT**

FROM table_list

Retrieve <u>specific columns</u> of all customers

select zipcode, fname, Iname from customer;

customer_num	fname	Iname	address	zipcode
101	Ludwig	Pauli	213 Erstwild Court	94086
102	Carole	Sadler	785 Geary St	94117
103	Philip	Currie	654 Poplar	94303

Customer Table

zipcode	fname	Iname
94086	Ludwig	Pauli
94117	Carole	Sadler
94303	Philip	Currie

3 rows selected

Can select in any order regardless of the order of the columns in the table. Data independence.

SELECT ... FROM clause

SELECT [DISTINCT] column_list

FROM table_list

■ Retrieve <u>distinct column values</u> from the table(s)

select distinct zipcode from customer;

OR

select unique zipcode from customer;

Example

Table data

zipcode 123456 654321

2 rows selected

13

SELECT ... FROM clause

SELECT [DISTINCT] column_list

FROM table_list

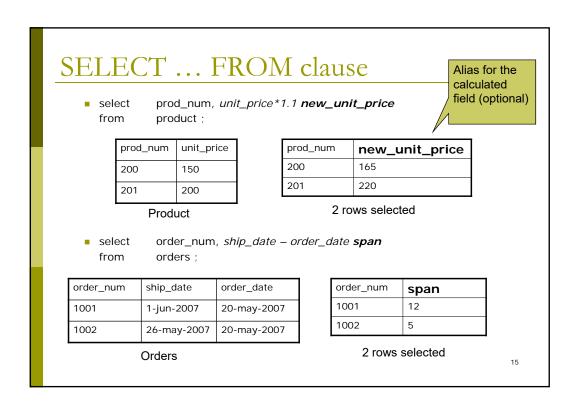
■ You may have <u>calculated (derived) columns</u> in the column_list:

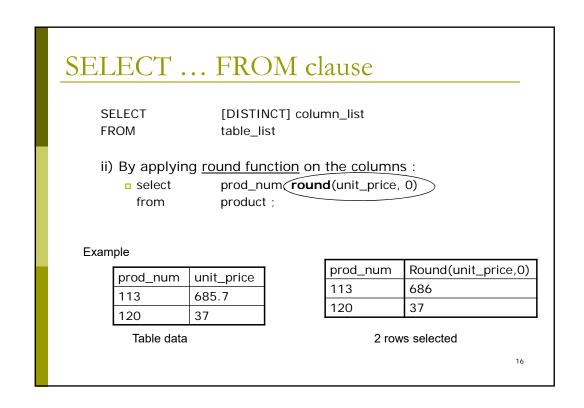
i) By performing <u>arithmetic operations</u> on the base table columns:

prod_num, *unit_price*1.1*) new_unit_price select from product;

select order_num, ship_date - order_date span from orders:

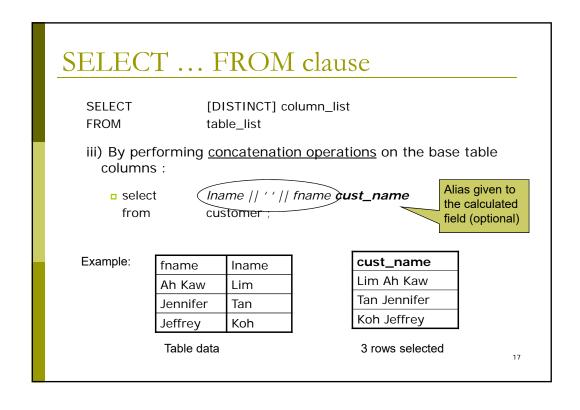
> You may give an alias to the calculated field (optional)

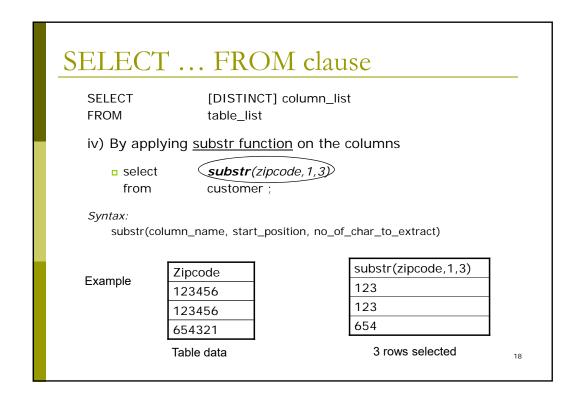




Unit 8a : SQL (Basic SELECT)

Page 8
AY2018/2019 S1





Unit 8a : SQL (Basic SELECT)

Page 9

AY2018/2019 S1

WHERE condition clause

- Row Selection, using the WHERE clause
 - To restrict the rows to be retrieved based on the condition(s) specified on the base table columns:

□ select prod_num, unit_price

product from

unit_price > 500; where

prod_num	unit_price	
113	685.5	
120	37	

prod_num	unit_price	
113	685.5	

1 row selected

Table data

WHERE condition clause

- Row Selection, using the WHERE clause
 - conditions can also be specifed on <u>derived</u> columns:

select order_num, ship_date - order_date span

from orders

where ship_date - order_date > 14;

order_num	ship_date	order_date	span
1004	30-may-2007	22-may-2007	8
1005	09-jun-2007	24-may-2007	16

order_num	span
1005	16

1 row selected

20

Table data

WHERE condition clause

Syntax:

[WHERE column_name < operator > value(s)]

- 5 basic search conditions that can be used in the WHERE clause:
 - comparison (=, <, >, <=, >=, <>)

Where salary > 5000Where state_code <> 'CA'

□ Range (BETWEEN, NOT BETWEEN)

Where salary BETWEEN 5000 and 10000 order_date BETWEEN '01-jul-1994' and Where

'31-jul-1994'

21

WHERE condition clause

Syntax :

[WHERE column_name < operator > value(s)]

Set membership (IN, NOT IN)

position IN ('Manager', 'Deputy Manager')

Pattern match (LIKE) with wildcards (%, _)

Where address LIKE 'Ang Mo Kio%'

Where state_code LIKE '_A'

□ Null (IS NULL, IS NOT NULL)

Where ship_instruct IS NULL

Compare with: where ship_instruct = '', any difference?

WHERE condition clause

■ Two or more conditions can be combined with AND / OR:

```
salary > 5000 AND position = 'Manager'
Where
Where
         order_date IS NULL OR ship_date IS NULL
```

23

ORDER BY clause

■ To sort the rows in the query result, in ascending or descending order of a column value or a combination of columns

```
Syntax:
```

[order by column_list [desc]]

where column_list :

- a column name in the select clause; or
- a column number (e.g. 1: the first element in the select clause, 2: the second element, and so on)
 - order by 1, 2 desc order by 1 desc, 2

24

ORDER BY clause

■ Examples :

Sort in descending order of ZIPCODE :

SELECT

FROM CUSTOMER

ORDER BY **ZIPCODE DESC**;

Sort in ascending order of LNAME

□ SELECT ZIPCODE, LNAME, FNAME

FROM CUSTOMER

ORDER BY 2;

Sort in ascending order of SUPPL_CODE, followed by descending order of UNIT_PRICE

SELECT

FROM PRODUCT

ORDER BY SUPPL_CODE, UNIT_PRICE DESC;

25

Summary

Basic SELECT statement

SELECT [DISTINCT] column_list

FROM table_name

{[INNER JOIN table_name ON condition]}

[WHERE condition]

[ORDER BY column_list [DESC]]

26

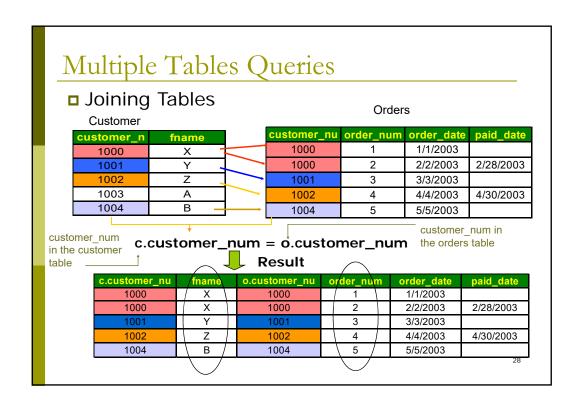
Multiple Tables Queries

- To obtain information from different tables (e.g. customer table, order table).
- Could use a <u>subquery</u> or a <u>join</u>.
- Example (List all the orders made by customers) :
 - Select fname, order_num

From customer c Inner Join orders o

On c.customer_num=o.customer_num;

27



Multiple Tables Queries

- To write a multiple table query :
 - Select c.customer_num, fname, order_num

From customer c

Inner Join orders o On c.customer_num = o.customer_num

- Include the table in the <u>FROM</u> clause
- Use the <u>INNER JOIN</u> clause to specify each additional table
- Include a <u>ON</u> clause to specify the column(s) to join, these columns must have compatible data types
- Whenever there is ambiguity in the source of the columns (same column name used in multiple tables), may use an alias for the table to qualify the column name

29

Multiple Tables Queries Examples

Example 1: List the customer's first name and name of the state they are in:

Select fname, state_name

From customer c

Inner Join state s On c.state_code = s.state_code

■ Example 2: List the order_num, order_date and the description of each product in the order 1002:

Select o.order_num, order_date, prod_desc

From orders o

Inner Join order_detail od On o.order_num = od.order_num
Inner Join product_desc pd On od.prod_num = pd.prod_num

Where o.order_num = 1002;

30

Unit 8a : SQL (Basic SELECT)

Page 15

AY2018/2019 S1

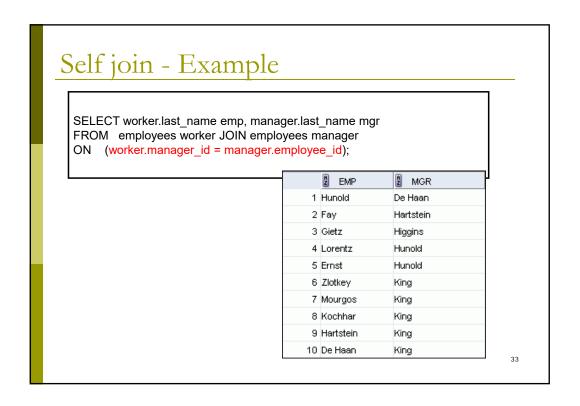
31

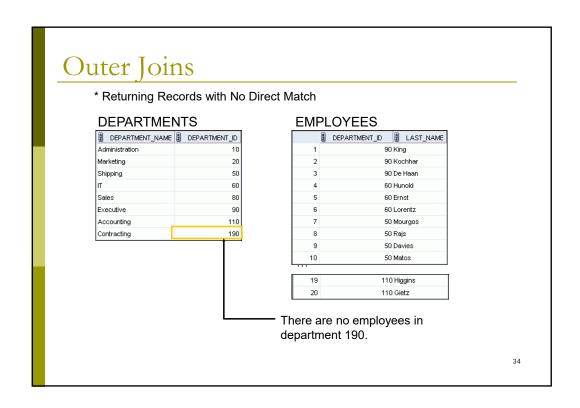
Multiple Tables Queries

- Various forms of JOINs available:
 - JOIN, CROSS JOIN, NATURAL JOIN
 - □ SELF JOIN
 - OUTER JOIN LEFT/ RIGHT/ FULL OUTER JOIN
- In particular, NATURAL JOIN can be used in place of INNER JOIN when the column names are identical in the joining tables. Example:
 - Select fname, order_numFrom customerNatural Join orders
- For the purpose of this module, we will concentrate on INNER JOIN.

Self Join **EMPLOYEES (MANAGER)** EMPLOYEES (WORKER) B EMPLOYEE_ID B LAST_NAME B MANAGER_ID B EMPLOYEE_ID LAST_NAME 100 King (null) 100 King 101 Kochhar 100 101 Kochhar 100 102 De Haan 102 De Haan 103 Hunold 102 103 Hunold 104 Ernst 104 Ernst 107 Lorentz 103 107 Lorentz 124 Mourgos 100 124 Mourgos 141 Rajs 124 141 Rajs 142 Davies 124 142 Davies 143 Matos 124 143 Matos MANAGER ID in the WORKER table is equal to EMPLOYEE_ID in the MANAGER table. 32

Unit 8a : SQL (Basic SELECT) Page 16 AY2018/2019 S1

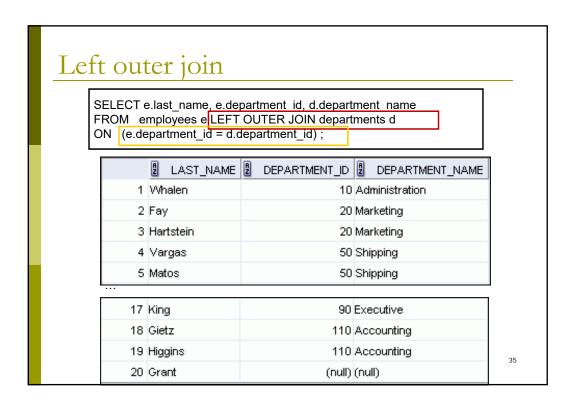


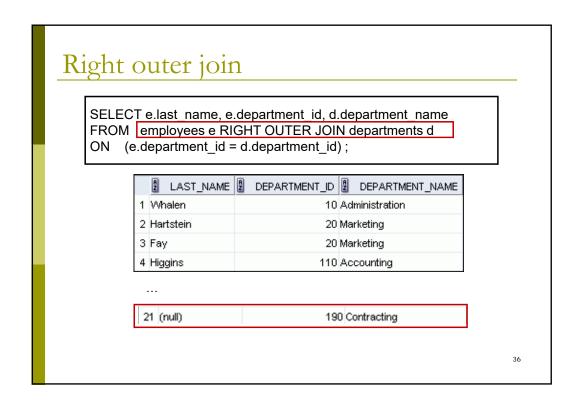


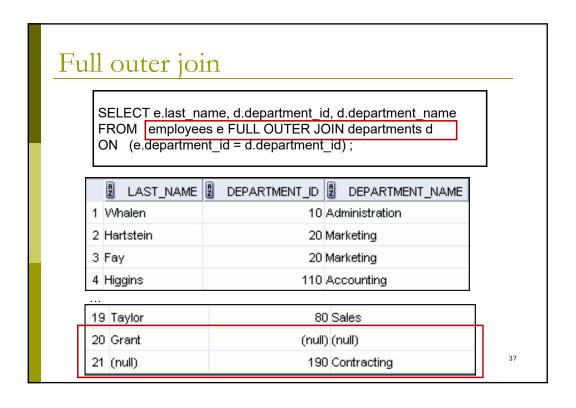
Unit 8a : SQL (Basic SELECT)

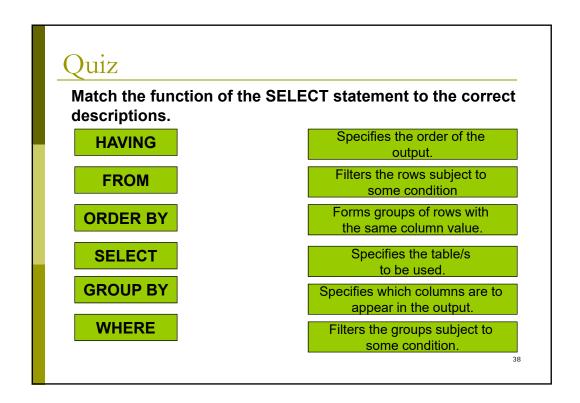
Page 17

AY2018/2019 S1









Unit 8a : SQL (Basic SELECT) Page 19 AY2018/2019 S1

Reference Materials, ELOs

□ Reference text: Database Systems, Connolly

■ DML : Ch 6 ■ DDL : Ch 7

□ ELOs :

- 'Evaluation Process of SELECT statement'
 - □ This ELO helps you to understand the evaluation process of a SELECT statement.
- 'Advanced_SELECT'
 - □ This ELO helps you to write the SELECT statement.

39

Unit 8a : SQL (Basic SELECT) AY2018/2019 S1