UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SECD 2523 DATABASE

LAB 4 : DML 3

NAME : TAHNUSSRI A/P MORTHY

MATRIC NUMBER : B23CS0071

SECTION : 08

LECTURER NAME : DR NOOR HIDAYAH ZAKARIA

Database Design Project

Oracle Baseball League Store Database

Project Scenario:

You are a small consulting company specializing in database development. You have just been awarded the contract to develop a data model for a database application system for a small retail store called Oracle Baseball League (OBL).

The Oracle Baseball League store serves the entire surrounding community selling baseball kit. The OBL has two types of customer, there are individuals who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts, and shorts. Additionally customers can represent a team when they purchase uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the inventory list, but teams get a discount on the list price depending on the number of players. When a customer places an order we record the order items for that order in our database.

OBL has a team of three sales representatives that officially only call on teams but have been known to handle individual customer complaints.

Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

Write SELECT Statements Using Data From Multiple Tables Using Equijoins and Non-Equijoins (S6L9 Objective 1)

In this exercise you will write SELECT statements to access data from more than one table.

Part 1: Creating Natural Joins.

1. Display all of the information about sales representatives and their addresses using a natural join.

```
SELECT * FROM sales_representatives
    NATURAL JOIN sales_rep_addresses;
```

Results | Explain | Describe | Saved SQL | History

ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52514
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

2. Adapt the query from the previous question to only show the id, first name, last name, address line 1, address line 2, city, email and phone_number for the sales representatives.

```
SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
    FROM sales_representatives
NATURAL JOIN sales_rep_addresses;
```

Results | Explain | Describe | Saved SQL | History

ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

Part 2: Creating Joins with the USING Clause

1. Adapt the previous query answer to use the USING clause instead of a natural join.

```
SELECT * FROM sales_representatives
    JOIN sales_rep_addresses sa
USING (id);
```

Results | Explain | Describe | Saved SQL | History

ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52514
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

2. Display all of the information about items and their price history by joining the items and price_history tables.

```
SELECT * FROM items
    JOIN price_history
USING (itm_number);
```

Results | Explain | Describe | Saved SQL | History

im01101045	under shirt	top worn under the game top	clothing	white	s	#010230125	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	#010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	#010230125	01/26/2017	01/26/2017	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	#010230126	02/12/2017	02/12/2017	7.99	-	-
im01101047	game top	team shirt with emblem	clothing	range	m	#010230127	04/25/2017	04/25/2017	24.99	-	-
im01101048	premium bat	high quality baseball bat	equipment	-	-	#010230128	05/31/2017	05/31/2017	149	-	-
im01101045	under shirt	top worn under the game top	clothing	white	s	#010230125	12/23/2023	12/23/2023	99.99	-	-

Part 3: Creating Joins with the ON Clause

1. Use an ON clause to join the customer and sales representative table so that you display the customer number, customer fist name, customer last name, customer phone number, customer email, sales representative id, sales representative first name, sales representative last name and sales representative email. You will need to use a table alias in your answer as both tables have columns with the same name.

```sql
SELECT
    c.ctr_number AS "Customer Number",
    c.first_name AS "Customer First Name",
    c.last_name AS "Customer Last Name",
    c.phone_number AS "Customer Phone Number",
    c.email AS "Customer Email",
    s.id AS "Sales Representative ID",
    s.first_name AS "Sales Representative First Name",
    s.last_name AS "Sales Representative Last Name",
    s.email AS "Sales Representative Email"
FROM
    customers c
JOIN
    sales_representatives s
ON
    c.sre_id = s.id;
```

Part 4- Creating Three-Way Joins with the ON Clause

1. Using the answer to Task 3 add a join that will allow the team name that the customer represents to be included in the results.

```sql
SELECT
    c.ctr_number AS "Customer Number",
    c.first_name AS "Customer First Name",
    c.last_name AS "Customer Last Name",
    c.phone_number AS "Customer Phone Number",
    c.email AS "Customer Email",
    s.id AS "Sales Representative ID",
    s.first_name AS "Sales Representative First Name",
    s.last_name AS "Sales Representative Last Name",
    s.email AS "Sales Representative Email",
    t.name AS "Team Name"
FROM
    customers c
JOIN
    sales_representatives s
ON
    c.sre_id = s.id
LEFT JOIN
    teams t
ON
    c.tem_id = t.id;
```

Part 5: Applying Additional Conditions to a Join

1. Using the answer to Task 4 add an additional condition to only show the results for the customer that has the number - c00001.

```sql
SELECT
    c.ctr_number AS "Customer Number",
    c.first_name AS "Customer First Name",
    c.last_name AS "Customer Last Name",
    c.phone_number AS "Customer Phone Number",
    c.email AS "Customer Email",
    s.id AS "Sales Representative ID",
    s.first_name AS "Sales Representative First Name",
    s.last_name AS "Sales Representative Last Name",
    s.email AS "Sales Representative Email",
    t.name AS "Team Name"
FROM
    customers c
JOIN
    sales_representatives s
ON
    c.sre_id = s.id
LEFT JOIN
    teams t
ON
    c.tem_id = t.id
WHERE
    c.ctr_number = 'c00001';
```

Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Representative ID	Sales Representative First Name	Sales Representative Last Name	Sales Representative Email	Team Name
c00001	Robert	Thornberry	0234567898	bob.thornberry@hotmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets

Part 6: Retrieving Records with Nonequijoins

1. Write a query that will display name and cost of the item with the number im01101045 on the 12th of December 2016. The output of the query should look like this:

The cost of the under shirt on this day was 14.99

```sql
SELECT 'The cost of the ' || i.name || ' on this day was ' || ph.price AS "cost of the item with the number im01101045 on the 12th of December "
    FROM items i
JOIN
    price_history ph
ON
    i.itm_number = ph.itm_number
WHERE
    i.itm_number = 'im01101045'
    AND ph.start_date = TO_DATE('2016-12-12', 'YYYY-MM-DD');
```

Explain Describe Saved SQL History

cost of the item with the number im01101045 on the 12th of December
The cost of the under shirt on this day was 14.99

Database Design Project

Oracle Baseball League Store Database

Project Scenario:

You are a small consulting company specializing in database development. You have just been awarded the contract to develop a data model for a database application system for a small retail store called Oracle Baseball League (OBL).

The Oracle Baseball League store serves the entire surrounding community selling baseball kit. The OBL has two types of customer, there are individuals who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts, and shorts. Additionally customers can represent a team when they purchase uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the inventory list, but teams get a discount on the list price depending on the number of players. When a customer places an order we record the order items for that order in our database.

OBL has a team of three sales representatives that officially only call on teams but have been known to handle individual customer complaints.

Section 6 Lesson 9 Exercise 2: Joining Tables Using JOIN

Write SELECT Statements Using Data From Multiple Tables Using Equijoins and Non-Equijoins (S6L9 Objective 1)

Part 1 : Use a Self-Join to Join a Table to Itself (S6L9 Objective 2)

1. Write a query that will display who the supervisor is for each of the sales representatives. The information should be displayed in two columns, the first column will be the first name and last name of the sales representative and the second will be the first name and last name of the supervisor. The column aliases should be Rep and Supervisor.

```sql
SELECT
    rep.first_name || ' ' || rep.last_name AS "Rep",
    sup.first_name || ' ' || sup.last_name AS "Supervisor"
FROM
    sales_representatives rep
JOIN
    sales_representatives sup
ON
    rep.supervisor_id = sup.id;
```

Rep	Supervisor
Charles Raymond	Charles Raymond
Victoria Wright	Charles Raymond
Barry Speed	Charles Raymond

Part 2 : Use OUTER joins (S6L9 Objective 3)

1. Write a query that will display all of the team and customer information even if there is no match with the tableon the left (team).

```sql
SELECT
    t.id AS "Team ID",
    t.name AS "Team Name",
    c.ctr_number AS "Customer Number",
    c.first_name AS "Customer First Name",
    c.last_name AS "Customer Last Name"
FROM
    teams t
LEFT JOIN
    customers c
ON
    t.id = c.tem_id;
```

Team ID	Team Name	Customer Number	Customer First Name	Customer Last Name
t001	Rockets	c00001	Robert	Thornberry
t002	Celtics	c00101	John	Doe
t003	Rovers	c01986	Maria	Galant
t004	Jets	-	-	-

Part 3 : Generating a Cartesian Product (S6L9 Objective 4)

1. Create a Cartesian product between the customer and sales representative tables.

```sql
SELECT
    c.ctr_number AS "Customer Number",
    c.first_name AS "Customer First Name",
    c.last_name AS "Customer Last Name",
    s.id AS "Sales Representative ID",
    s.first_name AS "Sales Representative First Name",
    s.last_name AS "Sales Representative Last Name"
FROM
    customers c
CROSS JOIN
    sales_representatives s;
```