



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**FACULTY OF COMPUTING**  
UTM Johor Bahru

**SECD2523 - DATABASE**

**SEMESTER I - SESSION 2023/2024**

**LECTURER : DR.IZYAN IZZATI BINTI KAMSANI**

**Lab Exercise SQL 4**

**SECTION : 06**

Name	Matric Number
EDWIN KOH WEI SHAN	A22EC5003

## 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.

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

- SELECT \*

FROM sales\_representatives NATURAL JOIN sales\_rep\_addresses;

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

Results											
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	DT52314	
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564	

- 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;

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

Results							
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

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

- `SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number  
FROM sales_representatives JOIN sales_rep_addresses  
USING (id);`

```
1 SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
2 FROM sales_representatives JOIN sales_rep_addresses
3 USING (id);
```

Results							
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

- 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);`

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

Results												
ITEM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME	
im0101045	under shirt	top worn under the game top	clothing	white	s	il010230125	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017	
im0101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017	
im0101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-	
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	11/09/2023	11/09/2023	99.99	-	-	
im0101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-	
im0101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-	
im0101044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	-	-	
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	14.9	11/09/2023	11/09/2023	

### 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 first 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.

— SELECT c.ctr\_number, c.first\_name, c.last\_name, c.phone\_number, c.email, s.id, s.first\_name, s.last\_name, s.email  
 FROM customers c JOIN sales\_representatives s  
 ON (c.sre\_id = s.id);

```
1 SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id, s.first_name, s.last_name, s.email
2 FROM customers c JOIN sales_representatives s
3 ON (c.sre_id = s.id);
```

Results	Explain	Describe	Saved SQL	History				
CTR_NUMBER	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ID	FIRST_NAME	LAST_NAME	EMAIL
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com

### 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.

— SELECT c.ctr\_number, c.first\_name AS "CTR\_FIRST\_NAME", c.last\_name AS "CTR\_LAST\_NAME",  
 c.phone\_number AS "CTR\_PHONE\_NUMBER", c.email AS "CTR\_EMAIL", s.id, s.first\_name AS "SALES\_FIRST\_NAME",  
 c.last\_name AS "SALES\_LAST\_NAME", s.email AS "SALES\_EMAIL", t.name AS "TEAM NAME"  
 FROM customers c JOIN sales\_representatives s  
 ON (c.sre\_id = s.id)  
 JOIN teams t  
 ON (c.team\_id = t.id);

Results	Explain	Describe	Saved SQL	History					
CTR_NUMBER	CTR_FIRST_NAME	CTR_LAST_NAME	CTR_PHONE_NUMBER	CTR_EMAIL	ID	SALES_FIRST_NAME	SALES_LAST_NAME	SALES_EMAIL	TEAM NAME
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics

## 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.

```
- SELECT c.ctr_number, c.first_name AS "CTR_FIRST_NAME", c.last_name AS "CTR_LAST_NAME",
c.phone_number AS "CTR_PHONE_NUMBER", c.email AS "CTR_EMAIL", s.id, s.first_name AS "SALES_FIRST_NAME",
c.last_name AS "SALES_LAST_NAME", s.email AS "SALES_EMAIL", t.name AS "TEAM NAME"
FROM customers c JOIN sales_representatives s
ON (c.sre_id = s.id)
JOIN teams t
ON (c.tem_id = t.id)
WHERE c.ctr_number = 'c00001';
```

```
1 SELECT c.ctr_number, c.first_name AS "CTR_FIRST_NAME", c.last_name AS "CTR_LAST_NAME", c.phone_number AS "CTR_PHONE_NUMBER", c.email AS "CTR_EMAIL", s.id,
2 s.first_name AS "SALES_FIRST_NAME", s.last_name AS "SALES_LAST_NAME", s.email AS "SALES_EMAIL", t.name AS "TEAM NAME"
3 FROM customers c JOIN sales_representatives s
4 ON (c.sre_id=s.id)
5 JOIN teams t
6 ON (c.tem_id=t.id)
7 WHERE c.ctr_number = 'c00001';
```

Results	Explain	Describe	Saved SQL	History					
CTR_NUMBER	CTR_FIRST_NAME	CTR_LAST_NAME	CTR_PHONE_NUMBER	CTR_EMAIL	ID	SALES_FIRST_NAME	SALES_LAST_NAME	SALES_EMAIL	TEAM NAME
c00001	Robert	Thornberry	01234567898	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 12<sup>th</sup> of December 2016. The output of the query should look like this:

The cost of the under shirt on this day was 14.99

```
- SELECT 'The cost of the ' || i.name || ' on this day was ' ||
p.price AS "Item Cost"
FROM items i JOIN price_history p
ON (i.item_number = p.item_number)
WHERE i.item_number = 'im01101045'
AND (TO_DATE ('16-Dec-2016', 'DD-MM-YYYY') BETWEEN p.start_date AND p.end_date);
```

```
1 SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "Price Cost"
2 FROM items i JOIN price_history p
3 ON (i.item_number=p.item_number)
4 WHERE i.item_number = 'im01101045'
5 AND (TO_DATE ('16-Dec-2016', 'DD-MM-YYYY') BETWEEN p.start_date AND p.end_date);
```

Results	Explain	Describe	Saved SQL	History
Price Cost				
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.

##### 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 table on the left (team).

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

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

## 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.

```
- SELECT r.first_name || ' ' || r.last_name AS "Rep", s.first_name  
|| ' ' || s.last_name AS "Supervisor"  
FROM salesRepresentatives r JOIN salesRepresentatives s  
ON (r.supervisor_id = s.id)
```

```
1 SELECT r.first_name || ' ' || r.last_name AS "Rep", s.first_name || ' ' || s.last_name AS "Supervisor"
2 FROM sales_representatives r JOIN sales_representatives s
3 ON (r.supervisor_id=s.id);
```

Results	Explain	Describe	Saved SQL	History
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 table on the left (team).

- SELECT \*

```
FROM teams t RIGHT OUTER JOIN customers c  
ON (t.id = c.team_id);
```

```
1 SELECT *  
2 FROM teams t RIGHT OUTER JOIN customers c  
3 ON (t.id=c.team_id);  
4
```

Results	Explain	Describe	Saved SQL	History									
ID	Name	Number_of_Players	Discount	Ctr_Number	Email	First_Name	Last_Name	Phone_Number	Current_Balance	Sre_Id	Tem_Id	Loyalty_Card_Number	
-	-	-	-	c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	Ic4587	
t001	Rockets	25	10	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	
-	-	-	-	c00012	Jjones@fremail.com	Jennifer	Jones	01505214598	0	-	-	Ic1015	
t002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	
-	-	-	-	c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	Ic2341	
t003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	

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

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

- *SELECT \**

*FROM customers*

*CROSS JOIN sales\_representatives;*

Results																
Explain Describe Saved SQL History																
CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	
c00012	jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c01986	margalit@delphiview.com	Maria	Galant	014427376589	125.65	sr03	t003	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	
c00012	jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds [Download](#)