NAME : JUDAH PAULO VIÑAS　　**INSTRUCTOR** : MS. JOCELYN LIPATA
PROGRAM, YEAR & SECTION : BSIT 3F　**COURSE** : IT 315 – ADVANCE DATABASE SYSTEMS

ACTIVITIES COMPILATION

ACTIVITY 1 : JOIN RESEARCH

1. What is an SQL **JOIN** command, and when do you need it?
   - An SQL JOIN command combines rows from two or more tables based on a common column. When you wish to retrieve data from many tables which have a common key or field, you need to use a JOIN.

2. How would you write a query to JOIN these two tables?
   - Assuming you have two tables, *Table1* and *Table2*, and you want to join them on a common column, say *id*, the SQL query might look like this:
       SELECT Table1.column1, Table2.column2
       FROM Table1
       JOIN Table2 ON Table1.id = Table2.id;

3. What types of JOINs are there?
   - **INNER JOIN:** Returns only rows where there is a match between both tables.
   - **LEFT (OUTER) JOIN:** Returns all rows from the left table and the corresponding rows from the right table. If no match is found, the right side returns NULL.
   - **RIGHT (OUTER) JOIN:** Returns all rows from the right table as well as the rows that match those in the left table. If no match is found, the left side returns NULL.
   - **FULL (OUTER) JOIN:** When a match is found in one of the tables, all rows are returned. If there is no match, NULL is returned in the table's columns.

4. What is an OUTER JOIN?
   - The **FULL OUTER JOIN** (aka **OUTER JOIN**) is used to return all of the records that have values in either the left or right table.

5. What is the difference between an SQL INNER JOIN and an SQL LEFT JOIN?
   - Use an **INNER JOIN** when you want to retrieve only the rows where there is a match in both tables based on the join condition. Use a **LEFT JOIN** when you want to retrieve all rows from the left table and the matched rows from the right table, with NULL values for non-matching rows.

6. What is the difference between a LEFT JOIN and a FULL JOIN?
   - **LEFT JOIN** returns only unmatched rows from the left table, as well as matched rows in both tables. **FULL OUTER JOIN** returns unmatched rows from both tables, as well as matched rows in both tables.

7. Write a query that will JOIN these two tables so that all rows from Table 1 are in the result.
   - Assuming you have *Table1* and *Table2* and want all rows from *Table1*, you would use a **LEFT JOIN**:
       SELECT Table1.column1, Table2.column2
       FROM Table1

>   ***LEFT JOIN Table2 ON Table1.id = Table2.id;***

8. How do you join more than two tables?
   - To join more than two tables, you can chain multiple JOIN clauses. For example:
     > ***SELECT Table1.column1, Table2.column2, Table3.column3***
     > ***FROM Table1 JOIN Table2 ON Table1.id = Table2.id***
     > ***JOIN Table3 ON Table2.id = Table3.id;***

9. How do you join a table to itself?
   - Joining a table to itself is called a self-join. You can do this using table names :
     > ***SELECT A.column1, B.column2***
     > ***FROM Table1 A***
     > ***JOIN Table1 B ON A.id = B.parent_id;***

10. Must the JOIN condition be equality?
   - No, the JOIN condition does not have to be equality. You can use other conditions, such as inequalities or ranges. For example:
     > ***SELECT A.column1, B.column2***
     > ***FROM Table1 A***
     > ***JOIN Table2 B ON A.column1 > B.column2;***

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* **COMPUTER STUDIES**

BS INFORMATION TECHNOLOGY	IT 315 - ADVANCE DATABASE SYSTEMS

ACTIVITY 2 : BASIC SQL QUERY

QUERY	OUTPUT

```
1    -- 1. Create a database named myDB
2  • CREATE DATABASE myDB;
```

▼ 🛢 mydb
   📑 Tables
   📑 Views
   📑 Stored Procedures
   📑 Functions

```
6    -- 2. Create a table named employee
7  • ⊖ CREATE TABLE employee (
8        emp_ID int PRIMARY KEY AUTO_INCREMENT,
9        first_name varchar(100),
10       last_name varchar(100),
11       hourly_pay decimal(10,2),
12       hire_date date
13     );
```

▼ 🛢 **mydb**
  ▼ 📑 Tables
    ▼ 📄 employee
      ▶ 📑 Columns
      ▶ 📑 Indexes
      ▶ 📑 Foreign Keys
      ▶ 📑 Triggers

```
15    -- 3. Add 10 records
16  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Judah Paulo", "Vinas", "100.00", "2024-08-2
17  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("John Mark David", "Pajenago", "100.00", "20
18  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Roland Joseph", "Cordez", "100.00", "2024-0
19  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Johnrick", "Babila", "100.00", "2024-08-29"
20  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("John Patrick", "Acbang", "100.00", "2024-0
21  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Dince", "Deliva", "100.00", "2024-08-29");
22  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Juan", "Lunatic", "50.00", "2024-08-29");
23  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Dave", "Manabat", "29.00", "2024-08-29");
24  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Jose", "Motte", "75.00", "2024-08-29");
25  • INSERT INTO employee (first_name, last_name, hourly_pay, hire_date) VALUE ("Dobido", "Baat", "90.00", "2024-08-29");
```

emp_ID	first_name	last_name	hourly_pay	hire_date
1	Judah Paulo	Vinas	100.00	2024-08-29
2	John Mark David	Pajenago	100.00	2024-08-29
3	Roland Joseph	Cordez	100.00	2024-08-29
4	Johnrick	Babila	100.00	2024-08-29
5	John Patrick	Acbang	100.00	2024-08-29
6	Dince	Deliva	100.00	2024-08-29
7	Juan	Lunatic	50.00	2024-08-29
8	Dave	Manabat	29.00	2024-08-29
9	Jose	Motte	75.00	2024-08-29
10	Dobido	Baat	90.00	2024-08-29
NULL	NULL	NULL	NULL	NULL

```
29    -- 4. Rename the table employee to employees
30  • ALTER TABLE employee RENAME employees;
```

▼ 🛢 **mydb**
  ▼ 📑 Tables
    ▼ 📄 employees
      ▶ 📑 Columns
      ▶ 📑 Indexes
      ▶ 📑 Foreign Keys
      ▶ 📑 Triggers

```
32    -- 5. Add phone_number column
33  • ALTER TABLE employees ADD phone_number varchar(11);
```

emp_ID	first_name	last_name	hourly_pay	hire_date	phone_number
1	Judah Paulo	Vinas	100.00	2024-08-29	NULL
2	John Mark David	Pajenago	100.00	2024-08-29	NULL
3	Roland Joseph	Cordez	100.00	2024-08-29	NULL
4	Johnrick	Babila	100.00	2024-08-29	NULL
5	John Patrick	Acbang	100.00	2024-08-29	NULL
6	Dince	Deliva	100.00	2024-08-29	NULL
7	Juan	Lunatic	50.00	2024-08-29	NULL
8	Dave	Manabat	29.00	2024-08-29	NULL
9	Jose	Motte	75.00	2024-08-29	NULL
10	Dobido	Baat	90.00	2024-08-29	NULL
NULL	NULL	NULL	NULL	NULL	NULL

```
34    -- 6. Change phone_number to email
35  • ALTER TABLE employees CHANGE phone_number email varchar(50);
```

emp_ID	first_name	last_name	hourly_pay	hire_date	email
1	Judah Paulo	Vinas	100.00	2024-08-29	NULL
2	John Mark David	Pajenago	100.00	2024-08-29	NULL
3	Roland Joseph	Cordez	100.00	2024-08-29	NULL
4	Johnrick	Babila	100.00	2024-08-29	NULL
5	John Patrick	Acbang	100.00	2024-08-29	NULL
6	Dince	Deliva	100.00	2024-08-29	NULL
7	Juan	Lunatic	50.00	2024-08-29	NULL
8	Dave	Manabat	29.00	2024-08-29	NULL
9	Jose	Motte	75.00	2024-08-29	NULL
10	Dobido	Baat	90.00	2024-08-29	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* **COMPUTER STUDIES**

| BS INFORMATION TECHNOLOGY | IT 315 - ADVANCE DATABASE SYSTEMS |

```sql
37      -- 7. Change the position of email next to last_name
38 •    ALTER TABLE employees MODIFY COLUMN email varchar(50) AFTER last_name;
```

emp_ID	first_name	last_name	email	hourly_pay	hire_date
1	Judah Paulo	Vinas	NULL	100.00	2024-08-29
2	John Mark David	Pajenago	NULL	100.00	2024-08-29
3	Roland Joseph	Cordez	NULL	100.00	2024-08-29
4	Johnrick	Babila	NULL	100.00	2024-08-29
5	John Patrick	Acbang	NULL	100.00	2024-08-29
6	Dince	Deliva	NULL	100.00	2024-08-29
7	Juan	Lunatic	NULL	50.00	2024-08-29
8	Dave	Manabat	NULL	29.00	2024-08-29
9	Jose	Motte	NULL	75.00	2024-08-29
10	Dobido	Baat	NULL	90.00	2024-08-29
NULL	NULL	NULL	NULL	NULL	NULL

```sql
40      -- 8. Add 2 records populate emp_ID, last name and first_name only
41 •    INSERT INTO employees (emp_ID, last_name, first_name) VALUE (11, "Skibidi","Cophal");
42 •    INSERT INTO employees (emp_ID, last_name, first_name) VALUE (12, "Sap","Neg-ah");
```

emp_ID	first_name	last_name	email	hourly_pay	hire_date
1	Judah Paulo	Vinas	NULL	100.00	2024-08-29
2	John Mark David	Pajenago	NULL	100.00	2024-08-29
3	Roland Joseph	Cordez	NULL	100.00	2024-08-29
4	Johnrick	Babila	NULL	100.00	2024-08-29
5	John Patrick	Acbang	NULL	100.00	2024-08-29
6	Dince	Deliva	NULL	100.00	2024-08-29
7	Juan	Lunatic	NULL	50.00	2024-08-29
8	Dave	Manabat	NULL	29.00	2024-08-29
9	Jose	Motte	NULL	75.00	2024-08-29
10	Dobido	Baat	NULL	90.00	2024-08-29
11	Cophal	Skibidi	NULL	NULL	NULL
12	Neg-ah	Sap	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

```sql
44      -- 9. Put a value on email column
45 •    UPDATE employees SET email = "juvinas@my.cspc.edu.ph" WHERE emp_ID = 1;
46 •    UPDATE employees SET email = "jopajenago@my.cspc.edu.ph" WHERE emp_ID = 2;
47 •    UPDATE employees SET email = "rocordez@my.cspc.edu.ph" WHERE emp_ID = 3;
48 •    UPDATE employees SET email = "jobabila@my.cspc.edu.ph" WHERE emp_ID = 4;
49 •    UPDATE employees SET email = "joacbang@my.cspc.edu.ph" WHERE emp_ID = 5;
50 •    UPDATE employees SET email = "dideliva@my.cspc.edu.ph" WHERE emp_ID = 6;
51 •    UPDATE employees SET email = "julunatic@my.cspc.edu.ph" WHERE emp_ID = 7;
52 •    UPDATE employees SET email = "damanabat@my.cspc.edu.ph" WHERE emp_ID = 8;
53 •    UPDATE employees SET email = "jomotte@my.cspc.edu.ph" WHERE emp_ID = 9;
54 •    UPDATE employees SET email = "dobaat@my.cspc.edu.ph" WHERE emp_ID = 10;
55 •    UPDATE employees SET email = "coskibidi@my.cspc.edu.ph" WHERE emp_ID = 11;
56 •    UPDATE employees SET email = "nesap@my.cspc.edu.ph" WHERE emp_ID = 12;
```

emp_ID	first_name	last_name	email	hourly_pay	hire_date
1	Judah Paulo	Vinas	juvinas@my.cspc.edu.ph	100.00	2024-08-29
2	John Mark David	Pajenago	jopajenago@my.cspc.edu.ph	100.00	2024-08-29
3	Roland Joseph	Cordez	rocordez@my.cspc.edu.ph	100.00	2024-08-29
4	Johnrick	Babila	jobabila@my.cspc.edu.ph	100.00	2024-08-29
5	John Patrick	Acbang	joacbang@my.cspc.edu.ph	100.00	2024-08-29
6	Dince	Deliva	dideliva@my.cspc.edu.ph	100.00	2024-08-29
7	Juan	Lunatic	julunatic@my.cspc.edu.ph	50.00	2024-08-29
8	Dave	Manabat	damanabat@my.cspc.edu.ph	29.00	2024-08-29
9	Jose	Motte	jomotte@my.cspc.edu.ph	75.00	2024-08-29
10	Dobido	Baat	dobaat@my.cspc.edu.ph	90.00	2024-08-29
11	Cophal	Skibidi	coskibidi@my.cspc.edu.ph	NULL	NULL
12	Neg-ah	Sap	nesap@my.cspc.edu.ph	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* **COMPUTER STUDIES**

| BS INFORMATION TECHNOLOGY | IT 315 - ADVANCE DATABASE SYSTEMS |

ACTIVITY 3 : LIBRARY TABLES

QUERY	OUTPUT

```
5      -- Create table author
6  ● ⊖ CREATE TABLE author (
7          id int PRIMARY KEY AUTO_INCREMENT,
8          author_name varchar(100),
9          birth_year int,
10         death_year int
11     );
```

```
12     -- Create table author
13  ● INSERT INTO author (author_name, birth_year) VALUE ("Marcella Cole", "1983");
14  ● INSERT INTO author (author_name, birth_year, death_year) VALUE ("Lisa Mullins", "1891", "1950");
15  ● INSERT INTO author (author_name, birth_year, death_year) VALUE ("Dennis Stokes", "1935", "1994");
16  ● INSERT INTO author (author_name, birth_year, death_year) VALUE ("Randolph Vasquez", "1957", "2004");
17  ● INSERT INTO author (author_name, birth_year, death_year) VALUE ("Daniel Branson", "1965", "1990");
```

id	author_name	birth_year	death_year
1	Marcella Cole	1983	NULL
2	Lisa Mullins	1891	1950
3	Dennis Stokes	1935	1994
4	Randolph Vasquez	1957	2004
5	Daniel Branson	1965	1990
NULL	NULL	NULL	NULL

```
21     -- Create table book
22  ● ⊖ CREATE TABLE book(
23         id int PRIMARY KEY AUTO_INCREMENT,
24         author_id int,
25         title varchar(100),
26         publish_year int,
27         publishing_house varchar(100),
28         rating decimal(10,2)
29     );
```

```
31     -- Insert values in table book
32  ● INSERT INTO book (title, publish_year, publishing_house, rating) VALUE ("Soulless", "2008", "Golden Albatros", "4.3");
33  ● INSERT INTO book (title, publish_year, publishing_house, rating) VALUE ("Weak Heart", "1980", "Diarmud Inc.", "3.8");
34  ● INSERT INTO book (author_id, title, publish_year, publishing_house, rating) VALUE ("4", "Faith Of Light", "1995", "Whi
35  ● INSERT INTO book (title, publish_year, publishing_house, rating) VALUE ("Memory Of Hope", "2000", "Rutis Enterprises",
36  ● INSERT INTO book (author_id, title, publish_year, publishing_house, rating) VALUE ("6", "Warrior Of Wind", "2005", "Ma
37  ● SELECT * FROM book;
```

id	author_id	title	publish_year	publishing_house	rating
1	NULL	Soulless	2008	Golden Albatros	4.30
2	NULL	Weak Heart	1980	Diarmud Inc.	3.80
3	4	Faith Of Light	1995	White Cloud Press	4.30
4	NULL	Memory Of Hope	2000	Rutis Enterprises	2.70
5	6	Warrior Of Wind	2005	Maverick	4.60
NULL	NULL	NULL	NULL	NULL	NULL

```
39     -- Create table adaptation
40  ● ⊖ CREATE TABLE adaptation(
41         book_id int PRIMARY KEY,
42         book_type varchar(100),
43         title varchar(100),
44         release_year int,
45         rating decimal(10,2)
46     );
```

```
48     -- Insert values in table adaptation
49  ● INSERT INTO adaptation (book_id, book_type, title, release_year, rating) VALUE ("1", "movie", "Gone With The Wolves: Tl
50  ● INSERT INTO adaptation (book_id, book_type, title, release_year, rating) VALUE ("3", "movie", "Companions Of Tomorrow",
51  ● INSERT INTO adaptation (book_id, book_type, title, release_year, rating) VALUE ("5", "movie", "Homeless Warrior", "200
52  ● INSERT INTO adaptation (book_id, book_type, title, release_year, rating) VALUE ("2", "movie", "Blacksmith With Silver"
53  ● INSERT INTO adaptation (book_id, book_type, title, release_year, rating) VALUE ("4", "movie", "Patrons And Bearers", "
54  ● SELECT * FROM adaptation;
```

book_id	book_type	title	release_year	rating
1	movie	Gone With The Wolves: The Beginning	2008	3.00
2	movie	Blacksmith With Silver	2014	4.30
3	movie	Companions Of Tomorrow	2001	4.20
4	movie	Patrons And Bearers	2004	3.20
5	movie	Homeless Warrior	2008	4.00
NULL	NULL	NULL	NULL	NULL

```
56     -- Create table book_review
57   ⊖ CREATE TABLE book_review(
58         book_id int,
59         review varchar(1000),
60         author varchar(100)
61     );
```

```
64     -- Insert values in table book_review
65   INSERT INTO book_review (book_id, review, author) VALUE ("1", "An incredible book", "Sylvia Jones");
66   INSERT INTO book_review (book_id, review, author) VALUE ("1", "Great, although it has some flaws", "Jessica Parker");
67   INSERT INTO book_review (book_id, review, author) VALUE ("2", "Dennis Strokes takes the reader for a ride full of emot:
68   INSERT INTO book_review (book_id, review, author) VALUE ("3", "Incredible craftsmanship of the author", "Martin Freema
69   INSERT INTO book_review (book_id, review, author) VALUE ("4", "Not the best book by this author", "Jude Falth");
70   INSERT INTO book_review (book_id, review, author) VALUE ("5", "Claudia Johnson at her best!", "Joe Marqiz");
71   INSERT INTO book_review (book_id, review, author) VALUE ("6", "I cannot recall more captivating plot", "Alexander Durh:
72   SELECT * FROM book_review;
```

book_id	review	author
1	An incredible book	Sylvia Jones
1	Great, although it has some flaws	Jessica Parker
2	Dennis Strokes takes the reader for a ride full o...	Thomas Green
3	Incredible craftsmanship of the author	Martin Freeman
4	Not the best book by this author	Jude Falth
5	Claudia Johnson at her best!	Joe Marqiz
6	I cannot recall more captivating plot	Alexander Durham

ACTIVITY 4 : JOIN

Table used :

author

id	author_name	birth_year	death_year
1	Marcella Cole	1983	NULL
2	Lisa Mullins	1891	1950
3	Dennis Stokes	1935	1994
4	Randolph Vasquez	1957	2004
5	Daniel Branson	1965	1990
NULL	NULL	NULL	NULL

book

id	author_id	title	publish_year	publishing_house	rating
1	NULL	Soulless	2008	Golden Albatros	4.30
2	NULL	Weak Heart	1980	Diarmud Inc.	3.80
3	4	Faith Of Light	1995	White Cloud Press	4.30
4	NULL	Memory Of Hope	2000	Rutis Enterprises	2.70
5	6	Warrior Of Wind	2005	Maverick	4.60
NULL	NULL	NULL	NULL	NULL	NULL

adaptation

book_id	book_type	title	release_year	rating
1	movie	Gone With The Wolves: The Beginning	2008	3.00
2	movie	Blacksmith With Silver	2014	4.30
3	movie	Companions Of Tomorrow	2001	4.20
4	movie	Patrons And Bearers	2004	3.20
5	movie	Homeless Warrior	2008	4.00
NULL	NULL	NULL	NULL	NULL

book_review

book_id	review	author
1	An incredible book	Sylvia Jones
1	Great, although it has some flaws	Jessica Parker
2	Dennis Strokes takes the reader for a ride full o...	Thomas Green
3	Incredible craftsmanship of the author	Martin Freeman
4	Not the best book by this author	Jude Falth
5	Claudia Johnson at her best!	Joe Marqiz
6	I cannot recall more captivating plot	Alexander Durham

QUERY	OUTPUT

```
72    -- 1. List All Books and Their Authors
73    --    Show the name of each author together with the title of the book they wrote and the
74    --    year in which that book was published.
75  • SELECT author_name, title, publish_year
76    FROM book
77    LEFT JOIN author
78    ON book.author_id = author.id;
```

author_name	title	publish_year
NULL	Soulless	2008
NULL	Weak Heart	1980
Randolph Vasquez	Faith Of Light	1995
NULL	Memory Of Hope	2000
NULL	Warrior Of Wind	2005

```
80    -- 2. List Authors and Books Published After 2005
81    -- Show the name of each author together with the title of the book they wrote
82    -- and the year in which that book was published. Show only books published after 2005.
83  • SELECT author_name, title, publish_year
84    FROM book
85    LEFT JOIN author
86    ON book.author_id = author.id
87    WHERE book.publish_year > 2005;
```

author_name	title	publish_year
NULL	Soulless	2008

```
89    -- 3. Show Books Adapted Within 4 Years and Rated Lower Than the Adaptation
90    --    For each book, show its title, adaptation title, adaptation year, and publication year.
91  • SELECT book.title AS book_title, adaptation.title AS adaptation_title, book.publish_year, adaptation.release_year
92    FROM book
93    JOIN adaptation
94    ON book.id = adaptation.book_id
95    WHERE (adaptation.release_year - book.publish_year) <= 4 AND adaptation.rating > book.rating;
```

book_title	adaptation_title	publish_year	release_year
Memory Of Hope	Patrons And Bearers	2000	2004

```
97    -- 4. Show All Books and Their Adaptations (If Any)
98    --    Show the title of each book together with the title of its adaptation and
99    --    the date of the release. Show all books, regardless of whether they had adaptations.
100 • SELECT book.title, adaptation.title, release_year
101   FROM book
102   LEFT JOIN adaptation
103   ON book.id = adaptation.book_id;
```

title	title	release_year
Soulless	Gone With The Wolves: The Beginning	2008
Weak Heart	Blacksmith With Silver	2014
Faith Of Light	Companions Of Tomorrow	2001
Memory Of Hope	Patrons And Bearers	2004
Warrior Of Wind	Homeless Warrior	2008

```
105   -- 5. Show All Books and Their Movie Adaptations
106   --    Show all books with their movie adaptations. Select each book's title,
107   --    the name of its publishing house, the title of its adaptation, and the type of
108   --    the adaptation. Keep the books with no adaptations in the result.
109 • SELECT book.title, book.publishing_house, adaptation.title, adaptation.book_type
110   FROM book
111   LEFT JOIN adaptation
112   ON book.id = adaptation.book_id
113   WHERE adaptation.book_type = 'movie' OR adaptation.book_type IS NULL;
```

title	publishing_house	title	book_type
Soulless	Golden Albatros	Gone With The Wolves: The Beginning	movie
Weak Heart	Diarmud Inc.	Blacksmith With Silver	movie
Faith Of Light	White Cloud Press	Companions Of Tomorrow	movie
Memory Of Hope	Rutis Enterprises	Patrons And Bearers	movie
Warrior Of Wind	Maverick	Homeless Warrior	movie

```sql
115    -- 6. Show All Books with Their Reviews (If Any)
116    --   Join the book_review and book tables using a RIGHT JOIN. Show the title
117    --   of the book, the corresponding review, and the name of the review's author.
118    --   Consider all books, even those that weren't reviewed.
119  • SELECT book.title, book_review.review, book_review.author
120    FROM book
121    RIGHT JOIN book_review
122    ON book.id = book_review.book_id;
```

title	review	author
Soulless	An incredible book	Sylvia Jones
Soulless	Great, although it has some flaws	Jessica Parker
Weak Heart	Dennis Strokes takes the reader for a ride full o…	Thomas Green
Faith Of Light	Incredible craftsmanship of the author	Martin Freeman
Memory Of Hope	Not the best book by this author	Jude Falth
Warrior Of Wind	Claudia Johnson at her best!	Joe Marqiz
NULL	I cannot recall more captivating plot	Alexander Durham

```sql
124    -- 7. List All the Books and All the Authors
125    --   Display the title of each book along with the name of its author.
126    --   Show all books, even those without an author. Show all authors,
127    --   even those who haven't published a book yet. Use a FULL JOIN.
128  • SELECT book.title, author.author_name
129    FROM book
130    LEFT JOIN author
131    ON book.author_id = author.id
132    UNION
133    SELECT book.title, author.author_name
134    FROM book
135    RIGHT JOIN author
136    ON book.author_id = author.id;
```

title	author_name
Soulless	NULL
Weak Heart	NULL
Faith Of Light	Randolph Vasquez
Memory Of Hope	NULL
Warrior Of Wind	NULL
NULL	Marcella Cole
NULL	Lisa Mullins
NULL	Dennis Stokes
NULL	Daniel Branson

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* COMPUTER STUDIES

| BS INFORMATION TECHNOLOGY | IT 315 - ADVANCE DATABASE SYSTEMS |

ACTIVITY 5 : VIEW

Table used :

employee

emp_num	emp_title	emp_lname	emp_fname	emp_bod	emp_hire_date	emp_area_code	emp_phone	park_code
100	Ms	Calderdale	Emma	1972-06-15	1992-03-15	101	324-9652	FR1001
101	Mr	Ricardo	Marshel	1978-03-19	1996-04-25	6731	623-1346	UK3452
102	Mr	Arshad	Arif	1969-11-14	1990-12-20	7253	675-8993	FR1001
103	Ms	Roberts	Anne	1974-10-16	1994-08-16	8127	135-4123	UK3452
104	Ms	Denver	Enrica	1980-11-08	2001-10-20	8427	985-2342	ZA1342
105	Ms	Namowa	Mirrelle	1998-03-14	2006-11-03	181	890-3243	FR1001
106	Ms	Smith	Gemma	1968-02-12	1989-01-05	1231	120-2311	ZA1342
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

tparkssold

park_code	park_name	park_city	park_country
FR1001	Fairyland	PARIS	FR
UK3452	PleasureLand	STOKE	UK
ZA1342	GoldTown	JOHANNESBURG	ZA

sales

park_name	line_price
FairyLand	15
FairyLand	140
GoldTown	12
GoldTown	115
PleasureLand	22
PleasureLand	168

QUERY	OUTPUT
25 -- E6.1 26 CREATE VIEW empfr AS 27 SELECT * 28 FROM employee 29 WHERE park_code = 'FR1001' 30 WITH CHECK OPTION;	<table><tr><th>emp_num</th><th>emp_title</th><th>emp_lname</th><th>emp_fname</th><th>emp_bod</th><th>emp_hire_date</th><th>emp_area_code</th><th>emp_phone</th><th>park_code</th></tr><tr><td>100</td><td>Ms</td><td>Calderdale</td><td>Emma</td><td>1972-06-15</td><td>1992-03-15</td><td>101</td><td>324-9652</td><td>FR1001</td></tr><tr><td>102</td><td>Mr</td><td>Arshad</td><td>Arif</td><td>1969-11-14</td><td>1990-12-20</td><td>7253</td><td>675-8993</td><td>FR1001</td></tr><tr><td>105</td><td>Ms</td><td>Namowa</td><td>Mirrelle</td><td>1998-03-14</td><td>2006-11-03</td><td>181</td><td>890-3243</td><td>FR1001</td></tr></table>
33 -- E6.2 34 UPDATE empfr 35 SET emp_phone = '324-9652' 36 WHERE emp_num = 100;	<table><tr><th>emp_num</th><th>emp_title</th><th>emp_lname</th><th>emp_fname</th><th>emp_bod</th><th>emp_hire_date</th><th>emp_area_code</th><th>emp_phone</th><th>park_code</th></tr><tr><td>100</td><td>Ms</td><td>Calderdale</td><td>Emma</td><td>1972-06-15</td><td>1992-03-15</td><td>101</td><td>324-9652</td><td>FR1001</td></tr><tr><td>102</td><td>Mr</td><td>Arshad</td><td>Arif</td><td>1969-11-14</td><td>1990-12-20</td><td>7253</td><td>675-8993</td><td>FR1001</td></tr><tr><td>105</td><td>Ms</td><td>Namowa</td><td>Mirrelle</td><td>1998-03-14</td><td>2006-11-03</td><td>181</td><td>890-3243</td><td>FR1001</td></tr></table>
40 -- E6.3 41 CREATE TABLE tparkssold (42 park_code varchar(10), 43 park_name varchar(35), 44 park_city varchar(50), 45 park_country char(2) 46);	

```
57    -- E6.4
58 •  CREATE VIEW emp_details AS
59    SELECT e.emp_num, e.park_code, p.park_name, e.emp_lname,
60            e.emp_fname, e.emp_hire_date, e.emp_bod
61    FROM employee e
62    LEFT JOIN tparkssold p
63    ON e.park_code = p.park_code;
```

emp_num	park_code	park_name	emp_lname	emp_fname	emp_hire_date	emp_bod
100	FR1001	Fairyland	Calderdale	Emma	1992-03-15	1972-06-15
101	UK3452	PleasureLand	Ricardo	Marshel	1996-04-25	1978-03-19
102	FR1001	Fairyland	Arshad	Arif	1990-12-20	1969-11-14
103	UK3452	PleasureLand	Roberts	Anne	1994-08-16	1974-10-16
104	ZA1342	GoldTown	Denver	Enrica	2001-10-20	1980-11-08
105	FR1001	Fairyland	Namowa	Mirrelle	2006-11-03	1998-03-14
106	ZA1342	GoldTown	Smith	Gemma	1989-01-05	1968-02-12

```
66    -- E6.5
67 •  SELECT emp_fname, emp_lname, park_name FROM emp_details;
```

emp_fname	emp_lname	park_name
Emma	Calderdale	Fairyland
Marshel	Ricardo	PleasureLand
Arif	Arshad	Fairyland
Anne	Roberts	PleasureLand
Enrica	Denver	GoldTown
Mirrelle	Namowa	Fairyland
Gemma	Smith	GoldTown

```
69    -- E6.6
70 •  CREATE TABLE sales (
71        park_name varchar(20),
72        line_price decimal(10.2)
73    );
```

```
89    -- E6.7
90 •  CREATE TABLE sale(
91        sale_date date
92    );
93 •  INSERT INTO sale(sale_date) VALUE ('2007-05-18');
94 •  select * from sale;
95
96 •  SELECT DISTINCT(DATE_FORMAT(sale_date, '%a-%e-%m-%y')) AS Date_Format
97    FROM sale;
```

Date_Format
Fri-18-05-07

```
98    -- E6.8
99 •  SELECT emp_fname, emp_lname,
100       CONCAT(DATE_FORMAT(emp_bod, '%m'), '01', UPPER(SUBSTR(emp_lname, 1, 6)))
101   AS user_id FROM employee;
102
```

emp_fname	emp_lname	user_id
Emma	Calderdale	0601CALDER
Marshel	Ricardo	0301RICARD
Arif	Arshad	1101ARSHAD
Anne	Roberts	1001ROBERT
Enrica	Denver	1101DENVER
Mirrelle	Namowa	0301NAMOWA
Gemma	Smith	0201SMITH

```
103   -- E6.9
104 • SELECT emp_fname, emp_lname, emp_bod
105   FROM employee WHERE DATE_FORMAT(emp_bod, '%d') = '14';
```

emp_fname	emp_lname	emp_bod
Arif	Arshad	1969-11-14
Mirrelle	Namowa	1998-03-14

```
107   -- E6.10
108 • SELECT emp_fname, emp_lname, CONCAT(SUBSTR(emp_phone, 1, 3), LOWER(SUBSTR(emp_fname, 1, 2)))
109   AS user_password FROM employee;
```

emp_fname	emp_lname	user_password
Emma	Calderdale	324em
Marshel	Ricardo	623ma
Arif	Arshad	675ar
Anne	Roberts	135an
Enrica	Denver	985en
Mirrelle	Namowa	890mi
Gemma	Smith	120ge

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* COMPUTER STUDIES

| BS INFORMATION TECHNOLOGY | IT 315 - ADVANCE DATABASE SYSTEMS |

ACTIVITY 6

Table used :

customer

	custid	custname	Car_number	Phone_number	Register_date
▶	P001	David Lim	WKM 1234	139845263	2012-06-14
	P002	Abu Hassan	DBA 8999	126377298	2023-05-14
	P003	Lw Ban Huat	PFL 3434	197894563	2005-01-14
	P004	Karigalan	BGN 2511	112356897	2023-05-13
	P007	Kamal Ibrahim	WEP 1103	145689471	2012-07-14
*	NULL	NULL	NULL	NULL	NULL

component_claim

	claimID	compID	quantity
▶	CL001	KK002	1
	CL001	KK005	2
	CL003	KK001	2
	CL012	KK003	1
	CL004	KK001	2
	CL002	KK003	1
	CL003	KK003	1

claim

	claimID	claim_date	custid
▶	CL001	2002-06-14	P0002
	CL002	2010-07-14	P0001
	CL003	2024-07-13	P0004
	CL004	2005-08-14	P0002
	CL012	2007-07-14	P0003
*	NULL	NULL	NULL

car_component

	compID	compName	price
▶	KK001	ISWARA Rear Lamp	100
	KK002	WIRA Front Mirror	450
	KK003	ISWARA Front Mirror	500
	KK004	WAJA Break Lamp	250
	KK005	WIRA Front Lamp	350
*	NULL	NULL	NULL

QUERY	OUTPUT
66 `-- 1.` 67 ● `SELECT MIN(price) AS MINPRICE` 68 `FROM car_component;`	MINPRICE ▶ 100
70 `-- 2.` 71 ● `SELECT MAX(price) AS MAXPRICE` 72 `FROM car_component;`	MAXPRICE ▶ 500
74 `-- 3.` 75 ● `SELECT AVG(price) AS AVGPRICE` 76 `FROM car_component;`	AVGPRICE ▶ 330.0000
78 `-- 4.` 79 ● `SELECT AVG(quantity) AS AVGQUANTITY` 80 `FROM component_claim;`	AVGQUANTITY ▶ 1.4286
82 `-- 5.` 83 ● `SELECT SUM(price)` 84 `FROM car_component;`	SUM(price) ▶ 1650
86 `-- 6.` 87 ● `SELECT SUM(quantity)` 88 `FROM component_claim;`	SUM(quantity) ▶ 10

Republic of the Philippines
Camarines Sur Polytechnic Colleges
Nabua, Camarines Sur
ISO 9001:2015

COLLEGE *of* COMPUTER STUDIES

BS INFORMATION TECHNOLOGY   IT 315 - ADVANCE DATABASE SYSTEMS

```
90      -- 7.
91  •   SELECT COUNT(custid)
92      FROM customer;
```

	COUNT(custid)
▶	5

```
94      -- 8.
95  •   SELECT COUNT(claimID)
96      FROM claim;
```

	COUNT(claimID)
▶	5

```
98      -- 9.
99  •   CREATE VIEW CUSTOMER_CLAIM AS
100     SELECT C.custid AS customer_id, C.custname AS name, C.Car_number, CL.claim_date
101     FROM customer C
102     JOIN claim CL
103     ON C.custid = CL.custid;
104 •   SELECT * FROM CUSTOMER_CLAIM;
```

```
108     -- 10.
109 •   CREATE VIEW car_claim AS
110     SELECT CC.compID, CC.compName AS company_name, CL.quantity, CC.price
111     FROM COMPONENT_CLAIM CL
112     JOIN CAR_COMPONENT CC
113     ON CL.compID = CC.compID;
114 •   SELECT * FROM car_claim;
115 •   drop view car_claim;
```

	compID	company_name	quantity	price
▶	KK002	WIRA Front Mirror	1	450
	KK005	WIRA Front Lamp	2	350
	KK001	ISWARA Rear Lamp	2	100
	KK003	ISWARA Front Mirror	1	500
	KK001	ISWARA Rear Lamp	2	100
	KK003	ISWARA Front Mirror	1	500
	KK003	ISWARA Front Mirror	1	500

ACTIVITY 7 : PROCEDURE

QUERY	OUTPUT
(see code block below)	

```sql
4        -- Create table Inventory
5  ●  ⊖ CREATE TABLE Inventory(
6            ItemID INT AUTO_INCREMENT PRIMARY KEY,
7            ItemName varchar(100),
8            Quantity int,
9            Price decimal(10,2)
10       );
```

```sql
13      -- Create stored procedure
14      DELIMITER //
15  ● ⊖ create procedure ManageInventory(
16          IN action varchar(10),
17          IN itemID int,
18          IN itemName varchar(100),
19          IN quantity int,
20          IN price decimal(10,2)
21      )
22   ⊖ BEGIN
23   ⊖     IF action="add" then
24              INSERT into Inventory(ItemName,Quantity,Price) Values (itemName, quantity, price);
25          ELSEIF action="update" then
26              UPDATE Inventory SET ItemName= itemName, Quantity = quantity, Price = price
27              WHERE ItemName = itemName;
28          ELSEIF action="retrieve" THEN
29              SELECT * FROM Inventory WHERE  ItemID = itemID;
30          ELSE
31              SELECT "Invalid Action";
32          END IF;
33      END//
34      DELIMITER ;
```

```sql
36  ●     -- Insert a value using the stored procedure
37          CALL ManageInventory("add", 1, "Lapis", 5, 50.00);
38  ●     SELECT * FROM Inventory;
```

ItemID	ItemName	Quantity	Price
1	Lapis	5	50.00
NULL	NULL	NULL	NULL