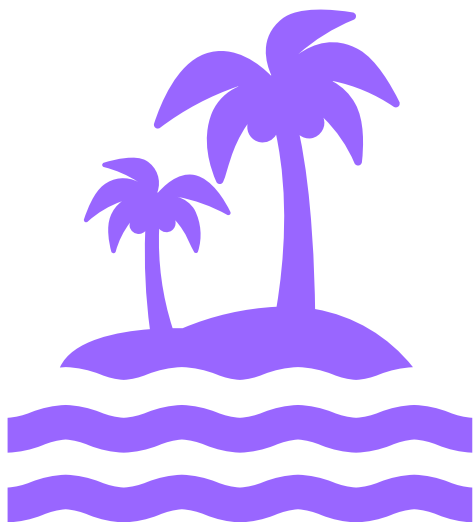


---

# ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ II

---

ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΑΞΙΔΙΩΤΙΚΟΥ ΓΡΑΦΕΙΟΥ



ΠΕΤΡΑΚΗ ΒΑΣΙΛΙΚΗ ΕΥΑΝΘΙΑ

19390193

ΕΡΓΑΣΤΗΡΙΑΚΟ ΤΜΗΜΑ 09 (ΠΕΜΠΤΗ 19:00 – 20:00)

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΜΕΡΟΣ Α</b> .....	2
CREATE, INSERT AND SELECT .....	2
Select.....	8
ENTITY – RELATIONSHIP DIAGRAM.....	10
VIEWS.....	11
Updatable.....	11
Non – updatable.....	12
BACKUP .....	13
<b>ΜΕΡΟΣ Β</b> .....	14
TRIGGERS .....	14
Insert .....	14
Update.....	16
Delete.....	18
FUNCTION .....	20
PROCEDURE .....	21

## ΜΕΡΟΣ Α

### CREATE, INSERT AND SELECT

---

```

1. DROP DATABASE IF EXISTS ETRAVEL;
2. CREATE DATABASE ETRAVEL;
3. USE ETRAVEL;
4. -- CREATING TABLE FOR THE WAYS OF TRANSPORTATION --
5. CREATE TABLE TRANSPORT(
6.     TID INT (2) NOT NULL AUTO_INCREMENT,
7.     TNAME VARCHAR (14) NOT NULL UNIQUE,
8.     PRIMARY KEY (TID)
9. );
10. -- INSERTING VALUES INTO TABLE TRANSPORT --
11. INSERT INTO TRANSPORT
12. VALUES(1, "AIRPLANE");
13. INSERT INTO TRANSPORT(TNAME)
14. VALUES("BUS"),
15.     ("TRAIN"),
16.     ("SHIP");
17. -- CREATING TABLE FOR CATEGORIES --
18. CREATE TABLE CATEGORIES(
19.     CID INT (2) NOT NULL AUTO_INCREMENT,
20.     CNAME VARCHAR (14) NOT NULL UNIQUE,
21.     PRIMARY KEY (CID)
22. );
23. -- INSERTING VALUES INTO TABLE CATEGORIES --
24. INSERT INTO CATEGORIES
25. VALUES(1, "WINTER");
26. INSERT INTO CATEGORIES (CNAME)
27. VALUES("ROMANTIC"),
28.     ("SUMMER"),
29.     ("FAMILY");
30. -- CREATING THE TABLE FOR THE TOURIST PACKAGES --
31. CREATE TABLE PACKAGES(

```

```

32.     PID INT (2) NOT NULL AUTO_INCREMENT,
33.     PLOC VARCHAR(30),
34.     BDATE DATE NOT NULL,
35.     EDATE DATE NOT NULL,
36.     CATEGORY VARCHAR(14) NOT NULL,
37.     TRANSP VARCHAR(14) NOT NULL,
38.     COST DECIMAL(10, 2),
39.     PRIMARY KEY (PID),
40.     FOREIGN KEY (TRANSP) REFERENCES TRANSPORT(TNAME),
41.     FOREIGN KEY (CATEGORY) REFERENCES CATEGORIES(CNAME)
42. );
43. INSERT INTO PACKAGES
44. VALUES (
45.     1,
46.     "VENICE",
47.     "2022-02-10",
48.     "2022-02-15",
49.     "ROMANTIC",
50.     "AIRPLANE",
51.     1100
52. );
53. INSERT INTO PACKAGES (PLOC, BDATE, EDATE, CATEGORY, TRANSP, COST)
54. VALUES (
55.     "BANSKO",
56.     "2021-12-19",
57.     "2021-12-21",
58.     "WINTER",
59.     "BUS",
60.     900
61. ),
62. (
63.     "PRAGUE",
64.     "2021-12-23",
65.     "2021-12-26",
66.     "FAMILY",
67.     "AIRPLANE",
68.     2000
69. ),
70. (
71.     "PALMA",
72.     "2022-06-10",
73.     "2022-06-30",
74.     "SUMMER",
75.     "AIRPLANE",
76.     7000
77. ),
78. (
79.     "PALMA",
80.     "2022-06-10",

```

```

81.         "2022-06-30",
82.         "SUMMER",
83.         "SHIP",
84.         4600
85.     ),
86.     (
87.         "PARIS",
88.         "2021-12-29",
89.         "2022-01-02",
90.         "ROMANTIC",
91.         "AIRPLANE",
92.         4000
93.     ),
94.     (
95.         "PARIS",
96.         "2021-12-29",
97.         "2022-01-02",
98.         "FAMILY",
99.         "AIRPLANE",
100.        6000
101.    ),
102.    (
103.        "ALPS",
104.        "2022-01-15",
105.        "2022-01-19",
106.        "WINTER",
107.        "AIRPLANE",
108.        3000
109.    ),
110.    (
111.        "ARACHOVA",
112.        "2022-01-20",
113.        "2022-01-23",
114.        "WINTER",
115.        "BUS",
116.        300
117.    );
118. -- CREATING THE TABLE FOR THE CUSTOMERS --
119. CREATE TABLE CUSTOMERS(
120.     CUSTID INT (3) NOT NULL AUTO_INCREMENT,
121.     CSNAME VARCHAR (50) NOT NULL,
122.     CADRESS VARCHAR (60) NOT NULL,
123.     CPHONE VARCHAR (11) NOT NULL,
124.     CEMAIL VARCHAR(255) NOT NULL,
125.     CPACK INT (2) NOT NULL,
126.     PRIMARY KEY(CUSTID),
127.     FOREIGN KEY(CPACK) REFERENCES PACKAGES(PID)
128. );
129. INSERT INTO CUSTOMERS

```

```

130. VALUES(
131.      1,
132.      "Alfonso Santana",
133.      "2137 Ipsum St.",
134.      "6999881132",
135.      "alfonsosantana@aol.com",
136.      4
137. );
138. INSERT INTO CUSTOMERS (CSNAME, CADRESS, CPHONE, CEMAIL, CPACK)
139. VALUES (
140.      "Constance Davidson",
141.      "9676 Cras Av.",
142.      "2226148298",
143.      "constancedavidson@outlook.com",
144.      7
145. ),
146. (
147.      "Leandra Beach",
148.      "1478 Nunc, Street",
149.      "2450974482",
150.      "leandrabeach@protonmail.com",
151.      6
152. ),
153. (
154.      "Chase Tyler",
155.      "Ap #379-6580 Auctor. Rd.",
156.      "6944467445",
157.      "chasetyler@yahoo.com",
158.      2
159. ),
160. (
161.      "Simone Harris",
162.      "Ap #723-9708 Vitae St.",
163.      "2628461581",
164.      "simoneharris@google.com",
165.      8
166. ),
167. (
168.      "Lars Conley",
169.      "Ap #220-6316 Odio. Road",
170.      "6922627264",
171.      "larsconley7819@google.com",
172.      5
173. ),
174. (
175.      "Latifah Hendricks",
176.      "9597 Turpis Avenue",
177.      "2932504105",
178.      "latifahhendricks@protonmail.com",

```

```

179.      1
180.    ),
181.    (
182.      "Dean Kirby",
183.      "862-5583 Egestas, Road",
184.      "2657681522",
185.      "deankirby@hotmail.com",
186.      3
187.    ),
188.    (
189.      "Olympia Morse",
190.      "Ap #126-6932 Libero. Road",
191.      "2778856521",
192.      "olympiamorse@google.com",
193.      3
194.    ),
195.    (
196.      "Jarrod Harmon",
197.      "Ap #350-9911 Elit St.",
198.      "6966751831",
199.      "jarrodharmon@hotmail.com",
200.      8
201.    ),
202.    (
203.      "Chloe Cline",
204.      "429-4974 Nascetur St.",
205.      "2775115683",
206.      "chloecline@yahoo.com",
207.      9
208.    ),
209.    ( "Olympia Morse",
210.      "Ap #126-6932 Libero. Road",
211.      "2778856521",
212.      "olympiamorse@google.com",
213.      4);
214. -- EVERY TABLE'S CONTENTS --
215. SELECT *
216. FROM CATEGORIES
217. ORDER BY CID ASC;
218. SELECT *
219. FROM TRANSPORT
220. ORDER BY TID ASC;
221. SELECT *
222. FROM PACKAGES
223. ORDER BY PID ASC;
224. SELECT *
225. FROM CUSTOMERS
226. ORDER BY CUSTID ASC;

```

	CID	CNAME
▶	1	WINTER
	2	ROMANTIC
	3	SUMMER
	4	FAMILY
•	NULL	NULL

	TID	TNAME
▶	1	AIRPLANE
	2	BUS
	3	TRAIN
	4	SHIP
•	NULL	NULL

	PID	PLOC	BDATE	EDATE	CATEGORY	TRANSP	COST
▶	1	VENICE	2022-02-10	2022-02-15	ROMANTIC	AIRPLANE	1100.00
	2	BANSKO	2021-12-19	2021-12-21	WINTER	BUS	900.00
	3	PRAGUE	2021-12-23	2021-12-26	FAMILY	AIRPLANE	2000.00
	4	PALMA	2022-06-10	2022-06-30	SUMMER	AIRPLANE	7000.00
	5	PALMA	2022-06-10	2022-06-30	SUMMER	SHIP	4600.00
	6	PARIS	2021-12-29	2022-01-02	ROMANTIC	AIRPLANE	4000.00
	7	PARIS	2021-12-29	2022-01-02	FAMILY	AIRPLANE	6000.00
	8	ALPS	2022-01-15	2022-01-19	WINTER	AIRPLANE	3000.00
	9	ARACHOVA	2022-01-20	2022-01-23	WINTER	BUS	300.00
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	CUSTID	CSNAME	CADRESS	CPHONE	CEMAIL	CPACK
▶	1	Alfonso Santana	2137 Ipsum St.	6999881132	alfonsosantana@aol.com	4
	2	Constance Davidson	9676 Cras Av.	2226148298	constancedavidson@outlook.com	7
	3	Leandra Beach	1478 Nunc, Street	2450974482	leandrabeach@protonmail.com	6
	4	Chase Tyler	Ap #379-6580 Auctor. Rd.	6944467445	chasetyler@yahoo.com	2
	5	Simone Harris	Ap #723-9708 Vitae St.	2628461581	simoneharris@google.com	8
	6	Lars Conley	Ap #220-6316 Odio. Road	6922627264	larsconley7819@google.com	5
	7	Latifah Hendricks	9597 Turpis Avenue	2932504105	latifahhendricks@protonmail.com	1
	8	Dean Kirby	862-5583 Egestas, Road	2657681522	deankirby@hotmail.com	3
	9	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	3
	10	Jarrold Harmon	Ap #350-9911 Elit St.	6966751831	jarroldharmon@hotmail.com	8
	11	Chloe Cline	429-4974 Nascetur St.	2775115683	chloeline@yahoo.com	9
	12	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	4
•	NULL	NULL	NULL	NULL	NULL	NULL



## Select

```

1.  -- ERWTHMA I --
2.  SELECT CATEGORIES.CID "CATEGORY ID",
3.         CATEGORIES.CNAME "CATEGORY NAME",
4.         COUNT(PACKAGES.CATEGORY) "NUMBER OF PACKAGES PER CATEGORY"
5.  FROM CATEGORIES,
6.         PACKAGES
7.  WHERE CATEGORIES.CNAME = PACKAGES.CATEGORY
8.  GROUP BY CATEGORIES.CID
9.  ORDER BY CATEGORIES.CID ASC;

```

	CATEGORY ID	CATEGORY NAME	NUMBER OF PACKAGES PER CATEGORY
▶	1	WINTER	3
	2	ROMANTIC	2
	3	SUMMER	2
	4	FAMILY	2

```

1.  -- ERWTHMA II --
2.  SELECT TRANSPORT.TID "TRANSPORT ID",
3.         TRANSPORT.TNAME "TRANSPORTATION METHODS",
4.         COUNT(PACKAGES TRANSP) "NUMBER OF PACKAGES PER TRANSPORTATION METHOD"
5.  FROM TRANSPORT,
6.         PACKAGES
7.  WHERE TRANSPORT.TNAME = PACKAGES TRANSP
8.  GROUP BY TRANSPORT.TID
9.  ORDER BY TRANSPORT.TID ASC;

```

	TRANSPORT ID	TRANSPORTATION METHODS	NUMBER OF PACKAGES PER TRANSPORTATION METHOD
▶	1	AIRPLANE	6
	2	BUS	2
	4	SHIP	1

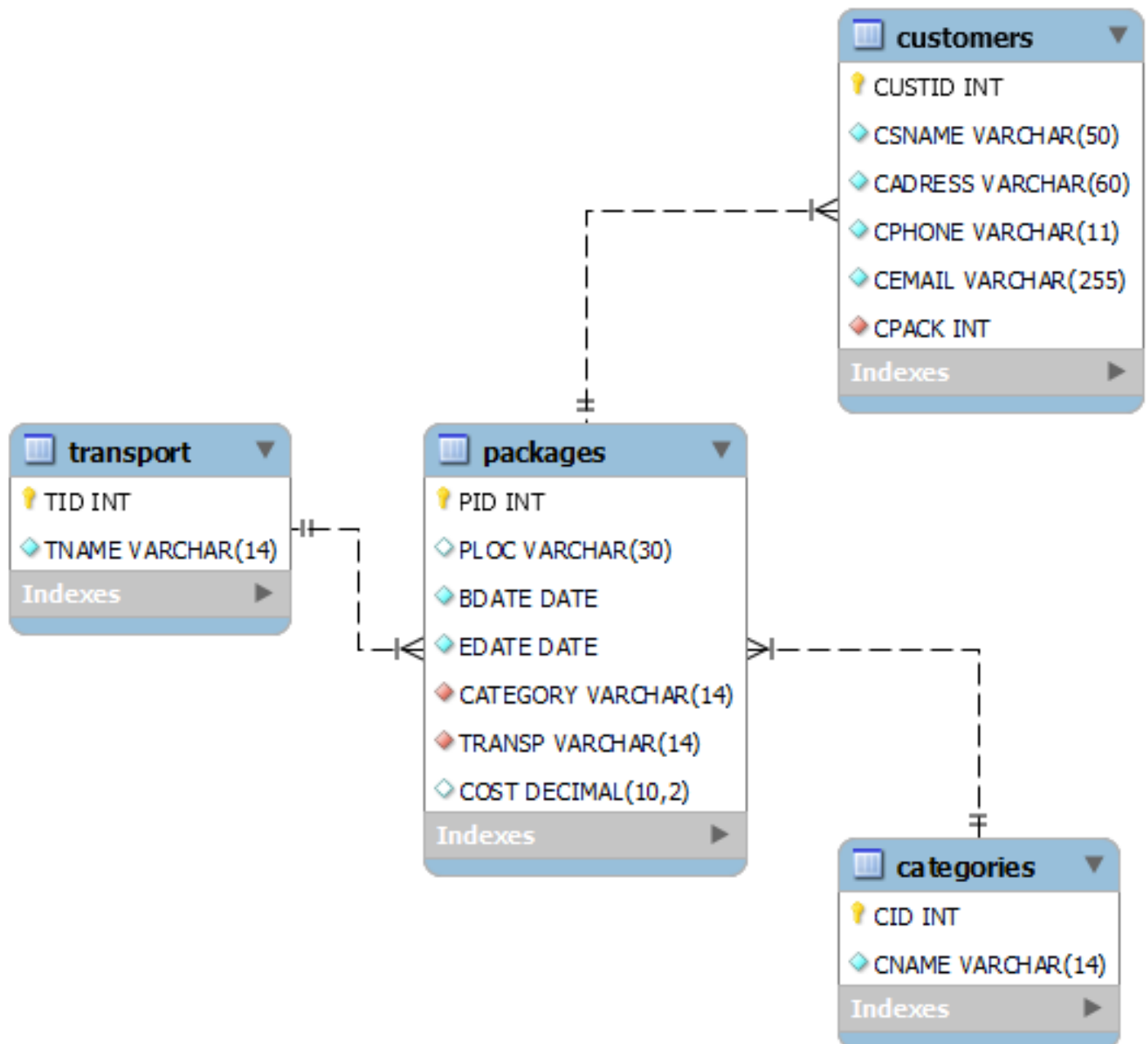
```

1.  -- ERWTHMA III --
2.  SELECT PACKAGES.PID "PACKAGE ID",
3.         CUSTOMERS.CSNAME "CUSTOMER NAME",
4.         SUM(COST) "TOTAL COST OF PACKAGE"
5.  FROM PACKAGES,
6.         CUSTOMERS
7.  WHERE PACKAGES.PID = CUSTOMERS.CPACK
8.  GROUP BY CUSTOMERS.CUSTID
9.  ORDER BY SUM(COST) DESC;

```

	PACKAGE ID	CUSTOMER NAME	TOTAL COST OF PACKAGE
►	4	Alfonso Santana	7000.00
	4	Olympia Morse	7000.00
	7	Constance Davidson	6000.00
	5	Lars Conley	4600.00
	6	Leandra Beach	4000.00
	8	Simone Harris	3000.00
	8	Jarrod Harmon	3000.00
	3	Dean Kirby	2000.00
	3	Olympia Morse	2000.00
	1	Latifah Hendricks	1100.00
	2	Chase Tyler	900.00
	9	Chloe Cline	300.00

## ENTITY – RELATIONSHIP DIAGRAM



## VIEWS

### Updatable

```

1.  -- CREATING AN UPDATABLE VIEW THAT SHOWS WHICH CUSTOMERS CHOSE THE PACKAGES TO
    PALMA --
2.  CREATE VIEW PALMA_VIEW (ID, NAME, ADRESS, PHONE, EMAIL, PACK) AS
3.  SELECT CUSTID,
4.         CSNAME,
5.         CADRESS,
6.         CPHONE,
7.         CEMAIL,
8.         CPACK
9.  FROM CUSTOMERS
10. WHERE (
11.         CPACK = 4
12.         OR CPACK = 5
13.     );
14. SELECT *
15. FROM PALMA_VIEW;

```

	ID	NAME	ADRESS	PHONE	EMAIL	PACK
►	1	Alfonso Santana	2137 Ipsum St.	6999881132	alfonsosantana@aol.com	4
	12	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	4
	6	Lars Conley	Ap #220-6316 Odio. Road	6922627264	larsconley7819@google.com	5

## Non – updatable

```

1.  -- CREATING A NON-UPDATABLE VIEW THAT SHOWS WHICH CUSTOMERS CHOSE THE ALPS PACKAGE
    --
2.  CREATE VIEW ALPS_VIEW (ID, NAME, ADRESS, PHONE, EMAIL, PACK) AS
3.  SELECT CUSTID,
4.         CSNAME,
5.         CADRESS,
6.         CPHONE,
7.         CEMAIL,
8.         CPACK
9.  FROM CUSTOMERS
10. WHERE (CPACK = 8)
11. GROUP BY CUSTID;
12. SELECT *
13. FROM ALPS_VIEW;

```

	ID	NAME	ADRESS	PHONE	EMAIL	PACK
►	5	Simone Harris	Ap #723-9708 Vitae St.	2628461581	simoneharris@google.com	8
	10	Jarrood Harmon	Ap #350-9911 Elit St.	6966751831	jarroodharmon@hotmail.com	8

## BACKUP

---



ETRAVELBACKUP.sql (Command Line)

## ΜΕΡΟΣ Β

### TRIGGERS

---

#### Insert

```

1. ALTER TABLE CATEGORIES
2. ADD NO_BOOKED_CAT INT;
3. -- UPDATE ROWS OF NEW COLUMN --
4. UPDATE CATEGORIES
5. SET
6.     NO_BOOKED_CAT = (SELECT
7.         COUNT(CATEGORY)
8.     FROM
9.         PACKAGES,
10.        CUSTOMERS
11.    WHERE
12.        PACKAGES.PID = CUSTOMERS.CPACK
13.        AND PACKAGES.CATEGORY = CATEGORIES.CNAME);
14. SELECT
15.     *
16. FROM
17.     CATEGORIES;
18. -- TRIGGER TO UPDATE BOOKED PACKAGES PER CATEGORY, IN CASE OF NEW INSERT --
19. DROP TRIGGER IF EXISTS CUST_INSERT;
20. DELIMITER //
21. CREATE TRIGGER CUST_INSERT
22. AFTER INSERT ON CUSTOMERS
23. FOR EACH ROW
24. BEGIN
25. UPDATE CATEGORIES
26. SET NO_BOOKED_CAT = (
27.     SELECT COUNT(CATEGORY)
28.     FROM PACKAGES,
29.         CUSTOMERS
30.     WHERE PACKAGES.PID = CUSTOMERS.CPACK
31.           AND PACKAGES.CATEGORY = CATEGORIES.CNAME
32. );
33. END;
34. //
35. DELIMITER ;
36. INSERT INTO CUSTOMERS (CSNAME, CADRESS, CPHONE, CEMAIL, CPACK)
37. VALUES (
38.     "Candace Graves",
39.     "1044 Mauris Road",

```

```

40.         "6995278527",
41.         "candacegraves1206@yahoo.com",
42.         2
43.     );
44. SELECT
45.     *
46. FROM
47.     CUSTOMERS;
48. SELECT
49.     *
50. FROM
51.     CATEGORIES;

```

	CUSTID	CSNAME	CADDRESS	CPHONE	CEMAIL	CPACK
►	1	Alfonso Santana	2137 Ipsum St.	6999881132	alfonsosantana@aol.com	4
	2	Constance Davidson	9676 Cras Av.	2226148298	constancedavidson@outlook.com	7
	3	Leandra Beach	1478 Nunc, Street	2450974482	leandrabeach@protonmail.com	6
	4	Chase Tyler	Ap #379-6580 Auctor. Rd.	6944467445	chasetyler@yahoo.com	2
	5	Simone Harris	Ap #723-9708 Vitae St.	2628461581	simoneharris@google.com	8
	6	Lars Conley	Ap #220-6316 Odio. Road	6922627264	larsconley7819@google.com	5
	7	Latifah Hendricks	9597 Turpis Avenue	2932504105	latifahhendricks@protonmail.com	1
	8	Dean Kirby	862-5583 Egestas, Road	2657681522	deankirby@hotmail.com	3
	9	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	3
	10	Jarrold Harmon	Ap #350-9911 Elit St.	6966751831	jarroldharmon@hotmail.com	8
	11	Chloe Cline	429-4974 Nascetur St.	2775115683	chloeline@yahoo.com	9
	12	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	4
	13	Candace Graves	1044 Mauris Road	6995278527	candacegraves1206@yahoo.com	2

	CID	CNAME	NO_BOOKED_CAT
►	1	WINTER	5
	2	ROMANTIC	2
	3	SUMMER	3
	4	FAMILY	3
•	NULL	NULL	NULL



## Update

```

1. DROP TRIGGER IF EXISTS CUST_UPDATE;
2. DELIMITER //
3. CREATE TRIGGER CUST_UPDATE
4. AFTER UPDATE ON CUSTOMERS
5. FOR EACH ROW
6. BEGIN
7. UPDATE CATEGORIES
8. SET NO_BOOKED_CAT = (
9.     SELECT COUNT(CATEGORY)
10.    FROM PACKAGES,
11.    CUSTOMERS
12.   WHERE PACKAGES.PID = CUSTOMERS.CPACK
13.     AND PACKAGES.CATEGORY = CATEGORIES.CNAME
14. );
15. END;
16. //
17. DELIMITER ;
18. UPDATE CUSTOMERS
19. SET
20.     CPACK = 6
21. WHERE
22.     CUSTID = 13;
23. SELECT
24.     *
25. FROM
26.     CUSTOMERS;
27. SELECT
28.     *
29. FROM
30.     CATEGORIES;

```

	CUSTID	CSNAME	CADRESS	CPHONE	CEMAIL	CPACK
▶	1	Alfonso Santana	2137 Ipsum St.	6999881132	alfonsosantana@aol.com	4
	2	Constance Davidson	9676 Cras Av.	2226148298	constancedavidson@outlook.com	7
	3	Leandra Beach	1478 Nunc, Street	2450974482	leandrabeach@protonmail.com	6
	4	Chase Tyler	Ap #379-6580 Auctor. Rd.	6944467445	chasetyler@yahoo.com	2
	5	Simone Harris	Ap #723-9708 Vitae St.	2628461581	simoneharris@google.com	8
	6	Lars Conley	Ap #220-6316 Odio. Road	6922627264	larsconley7819@google.com	5
	7	Latifah Hendricks	9597 Turpis Avenue	2932504105	latifahhendricks@protonmail.com	1
	8	Dean Kirby	862-5583 Egestas, Road	2657681522	deankirby@hotmail.com	3
	9	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	3
	10	Jarrod Harmon	Ap #350-9911 Elit St.	6966751831	jarrodharmon@hotmail.com	8
	11	Chloe Cline	429-4974 Nascetur St.	2775115683	chloedine@yahoo.com	9
	12	Olympia Morse	Ap #126-6932 Libero. Road	2778856521	olympiamorse@google.com	4
	13	Candace Graves	1044 Mauris Road	6995278527	candacegraves1206@yahoo.com	6

	CID	CNAME	NO_BOOKED_CAT
▶	1	WINTER	4
	2	ROMANTIC	3
	3	SUMMER	3
	4	FAMILY	3
✱	NULL	NULL	NULL

## Delete

```

1. DROP TRIGGER IF EXISTS CUST_DELETE;
2. DELIMITER //
3. CREATE TRIGGER CUST_DELETE
4. AFTER DELETE ON CUSTOMERS
5. FOR EACH ROW
6. BEGIN
7. UPDATE CATEGORIES
8. SET NO_BOOKED_CAT = (
9.     SELECT COUNT(CATEGORY)
10.    FROM PACKAGES,
11.    CUSTOMERS
12.   WHERE PACKAGES.PID = CUSTOMERS.CPACK
13.     AND PACKAGES.CATEGORY = CATEGORIES.CNAME
14. );
15. END;
16. //
17. DELIMITER ;
18. DELETE FROM CUSTOMERS
19. WHERE
20.     CUSTID = 13;
21. SELECT
22.     *
23. FROM
24.     CUSTOMERS;
25. SELECT
26.     *
27. FROM
28.     CATEGORIES;

```

	CUSTID	CSNAME	CADDRESS	CPHONE	CEMAIL	CPACK
►	1	Alfonso Santana	2137 Ipsum St.	6999881132	alfonsosantana@aol.com	4
	2	Constance Davidson	9676 Cras Av.	2226148298	constancedavidson@outlook.com	7
	3	Leandra Beach	1478 Nunc, Street	2450974482	leandrabeach@protonmail.com	6
	4	Chase Tyler	Ap #379-6580 Auctor, Rd.	6944467445	chasetyler@yahoo.com	2
	5	Simone Harris	Ap #723-9708 Vitae St.	2628461581	simoneharris@google.com	8
	6	Lars Conley	Ap #220-6316 Odio, Road	6922627264	larsconley7819@google.com	5
	7	Latifah Hendricks	9597 Turpis Avenue	2932504105	latifahhendricks@protonmail.com	1
	8	Dean Kirby	862-5583 Egestas, Road	2657681522	deankirby@hotmail.com	3
	9	Olympia Morse	Ap #126-6932 Libero, Road	2778856521	olympiamorse@google.com	3
	10	Jarrold Harmon	Ap #350-9911 Elit St.	6966751831	jarroldharmon@hotmail.com	8
	11	Chloe Cline	429-4974 Nascetur St.	2775115683	chloedine@yahoo.com	9
	12	Olympia Morse	Ap #126-6932 Libero, Road	2778856521	olympiamorse@google.com	4
•	NULL	NULL	NULL	NULL	NULL	NULL

	CID	CNAME	NO_BOOKED_CAT
►	1	WINTER	4
	2	ROMANTIC	2
	3	SUMMER	3
	4	FAMILY	3
•	NULL	NULL	NULL

## FUNCTION

```

1. DROP FUNCTION IF EXISTS PACKCOST;
2. DELIMITER !
3. CREATE FUNCTION PACKCOST(CSNAME VARCHAR(50))
4. RETURNS DECIMAL DETERMINISTIC
5. BEGIN
6. DECLARE CUSTCOST DECIMAL(10, 2);
7. SELECT
8.     SUM(COST)
9. FROM
10.    PACKAGES,
11.    CUSTOMERS
12. WHERE
13.     PACKAGES.PID = CUSTOMERS.CPACK
14.     AND CUSTOMERS.CSNAME = CSNAME
15. GROUP BY CUSTOMERS.CSNAME INTO CUSTCOST;
16. RETURN CUSTCOST;
17. END !
18. DELIMITER ;
19. SELECT PACKCOST("Olympia Morse") AS 'TOTAL SUM FOR CHOSEN CUSTOMER';
    
```

	TOTAL SUM FOR CHOSEN CUSTOMER
▶	9000

## PROCEDURE

```

1. DROP PROCEDURE IF EXISTS CUSTPROC;
2. DELIMITER $$
3. CREATE PROCEDURE CUSTPROC(CEMAIL VARCHAR(255))
4. BEGIN
5. SELECT COUNT(CPACK) "TOTAL NUMBER OF PACKAGES BOOKED"
6. FROM CUSTOMERS
7. WHERE CUSTOMERS.CEMAIL = CEMAIL;
8. SELECT
9.     DATEDIFF(PACKAGES.EDATE, PACKAGES.BDATE) AS DateDiff
10. FROM
11.     PACKAGES,
12.     CUSTOMERS
13. WHERE
14.     CUSTOMERS.CEMAIL = CEMAIL
15.     AND CUSTOMERS.CPACK = PACKAGES.PID;
16. END $$
17. DELIMITER ;
18. CALL CUSTPROC("chasetyler@yahoo.com");
19. CALL CUSTPROC("olympiamorse@google.com");
    
```

	TOTAL NUMBER OF PACKAGES BOOKED
▶	1

	DateDiff
▶	2

	TOTAL NUMBER OF PACKAGES BOOKED
▶	2

	DateDiff
▶	3
	20