Nama : Raden Sadiah Maharani

NIM : 2107126368

Basis Data Lanjut

Praktikum 8

1. Tuliskan query dengan perintah menggunakan penyataan INSERT eksplisit untuk menambah empat CD baru pada DJ on Demand ke tabel copy_d_cds. Setelah menyelesaikan entri, jalankan pernyataan SELECT * untuk memverifikasi pekerjaan Anda.

→

```
CREATE TABLE copy_d_cds
AS SELECT * FROM d_cds;
```

```
INSERT INTO copy_d_cds(cd_number, title, producer, year) VALUES (97, 'Celebrate the Day', 'R & B Inc.', 2003);
```

```
INSERT INTO copy_d_cds(cd_number, title, producer, year) VALUES (98, 'Holiday Tunes for All Ages', 'Tunes are Us', 2004);
```

```
INSERT INTO copy_d_cds(cd_number, title, producer, year) VALUES (99, 'Party Music', 'Old Town Records', 2004);
```

```
INSERT INTO copy_d_cds(cd_number, title, producer, year) VALUES (100, 'Best of Rock and Roll', 'Old Town Records', 2004);
```

```
SELECT * FROM copy_d_cds;
```

Result :

CD_NUMBER	TITLE	PRODUCER	YEAR
90	The Celebrants Live in Concert	Old Town Records	1997
91	Party Music for All Occasions	The Music Man	2000
92	Back to the Shire	Middle Earth Records	2002
93	Songs from My Childhood	Old Town Records	1999
94	Carpe Diem	R & B Inc.	2000
95	Here Comes the Bride	The Music Man	2001
96	Graduation Songbook	Tunes Are Us	1998
98	Whirled Peas	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004
97	Celebrate the Day	R & B Inc.	2003
More than 10 rows available. Increase rows selector to view more rows.			

2. DJ on Demand memiliki dua acara baru yang akan datang. Satu acara adalah pesta sepak bola musim gugur dan acara lainnya adalah pesta bertema tahun enam puluhan. Klien DJ on Demand meminta lagu yang ditampilkan di tabel untuk acara mereka. Tambahkan lagu-lagu ini ke tabel copy_d_songs menggunakan pernyataan INSERT implisit.

→
```
CREATE TABLE copy_d_songs AS
SELECT * FROM d_songs;


INSERT INTO copy_d_songs VALUES(52,'Surfing
Summer',NULL,NULL,12);


INSERT INTO copy_d_songs VALUES(53,'Victory
Victory','5 min',NULL,12);


SELECT * FROM copy_d_songs;
```

Result :

ID	TITLE	DURATION	ARTIST	TYPE_CODE
45	Its Finally Over	5 min	The Hobbits	12
46	Im Going to Miss My Teacher	2 min	Jane Pop	12
47	Hurrah for Today	3 min	The Jubilant Trio	77
48	Meet Me At the Altar	6 min	Bobby West	1
49	Lets Celebrate	8 min	The Celebrants	77
50	All These Years	10 min	Diana Crooner	88
52	Surfing Summer	-	-	12
53	Victory Victory	5 min	-	12

3. Buat tabel bernama rep_email menggunakan pernyataan berikut:

```
first_name VARCHAR2(10),

last_name VARCHAR2(10),

email_address VARCHAR2(10))
```

Isi tabel ini dengan menjalankan kueri pada tabel karyawan yang hanya menyertakan karyawan yang merupakan REP.

→

```
ALTER TABLE rep_email DROP column id;


ALTER TABLE rep_email ADD id NUMBER(6,0)
CONSTRAINT rel_id_pk PRIMARY KEY;


INSERT INTO rep_email(id, first_name, last_name,
email_address) SELECT employee_id, first_name,
last_name, email

FROM employees

WHERE job_id LIKE '%\_REP' ESCAPE '\';

SELECT * FROM rep_email;
```

Result :

FIRST_NAME	LAST_NAME	EMAIL_ADDRESS	ID
Ellen	Abel	EABEL	174
Jonathon	Taylor	JTAYLOR	176
Kimberely	Grant	KGRANT	178
Pat	Fay	PFAY	202

4. Buat perubahan harga pada tabel copy_f_food_items. Dengan perubahan harga strawberry shake akan dinaikkan dari $3,59 menjadi $3,75, dan harga kentang goreng akan naik menjadi $1,20.

→ `CREATE TABLE copy_f_food_items AS ( SELECT * FROM f_food_items);`

```
UPDATE copy_f_food_items SET price = 3.75 WHERE
LOWER(description) = 'strawberry shake';

UPDATE copy_f_food_items SET price = 1.20 WHERE
LOWER(description) = 'fries';


SELECT * FROM copy_f_food_items;
```

Result :

FOOD_ITEM_NUMBER	DESCRIPTION	PRICE	REGULAR_CODE	PROMO_CODE
90	Fries	1.2	20	-
93	Strawberry Shake	3.75	-	110

5. Tambahkan pesanan yang ditampilkan ke tabel copy_f_orders Makanan Cepat Saji Global:

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5680	June 12, 2004	159.78	145	9
5691	09-23-2004	145.98	225	12
5701	July 4, 2004	229.31	230	12

→ `CREATE TABLE copy_f_orders`

```
AS ( SELECT * FROM f_orders);


INSERT INTO
copy_f_orders(order_number,order_date,order_total
,cust_id,staff_id) VALUES(5680,TO_DATE('June 12,
2004','fmMonth dd, yyyy'),159.78,145,9);

INSERT INTO
copy_f_orders(order_number,order_date,order_total
,cust_id,staff_id) VALUES(5691,TO_DATE('09-23-
2004','mm-dd-yyyy'),145.98,225,12);

INSERT INTO
copy_f_orders(order_number,order_date,order_total
,cust_id,staff_id) VALUES(5701,TO_DATE('July 4,
2004','fmMonth dd, yyyy'),229.31,230,12);


SELECT * FROM copy_f_orders;
```

Result :

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5678	12/10/2002	103.02	123	12
5680	06/12/2004	159.78	145	9
5701	07/04/2004	229.31	230	12
5691	09/23/2004	145.98	225	12

6. Sue Doe telah menjadi anggota staf Global Foods yang luar biasa dan telah mendapatkan kenaikan gaji. Dia sekarang akan dibayar sama dengan Bob Miller. Perbarui catatannya di copy_f_staffs.

→
```
CREATE TABLE copy_f_staffs AS ( SELECT * FROM
f_staffs);
```

```
UPDATE copy_f_staffs SET salary = (SELECT salary
    FROM copy_f_staffs WHERE LOWER(first_name ||
    ' ' || last_name) = 'bob miller')
WHERE LOWER(first_name || ' ' || last_name) = 'sue
    doe';


SELECT * from copy_f_staffs;
```

Result :

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	OVERTIME_RATE	TRAINING	STAFF_TYPE	MANAGER_ID	MANAGER_BUDGET	MANAGER_TARGET
12	Sue	Doe	07/01/1980	10	10.25	-	Order Taker	19	-	-
9	Bob	Miller	03/19/1979	10	-	Grill	Cook	19	-	-
19	Monique	Tuttle	03/30/1969	60	-	-	Manager	-	50000	70000

7. Sekarang semua informasi tersedia untuk Kai Kim, perbarui catatan Global Fast Foods-nya untuk menyertakan yang berikut: Kai akan memiliki manajer yang sama dengan Sue Doe. Dia tidak memenuhi syarat untuk lembur. Biarkan nilai untuk pelatihan, anggaran manajer, dan target manajer sebagai null.

→
```
INSERT INTO
copy_f_staffs(id, first_name, last_name,
    birthdate, salary, overtime_rate, training,
    staff_type, manager_id, manager_budget,
    manager_target)
VALUES (25, 'Kai', 'Kim', TO_DATE('03-Nov-
    1988','fmdd-Mon-yyyy'), 6.75, NULL, NULL,
    'Order Taker', NULL, NULL, NULL);


UPDATE copy_f_staffs SET manager_id = (SELECT
    manager_id FROM copy_f_staffs WHERE
    LOWER(first_name || ' ' || last_name) = 'sue
    doe')
```

```
WHERE LOWER(first_name || ' ' || last_name) = 'kai
      kim';


SELECT * FROM copy_f_staffs;
```

Result :

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	OVERTIME_RATE	TRAINING	STAFF_TYPE	MANAGER_ID	MANAGER_BUDGET	MANAGER_TARGET
12	Sue	Doe	07/01/1980	10	10.25	-	Order Taker	19	-	-
9	Bob	Miller	03/19/1979	10	-	Grill	Cook	19	-	-
19	Monique	Tuttle	03/30/1969	60	-	-	Manager	-	50000	70000
25	Kai	Kim	11/03/1988	6.75	-	-	Order Taker	19	-	-

8. Buat salinan tabel karyawan dan beri nama lesson7_emp;

   Setelah tabel ini ada, tulis pernyataan hapus berkorelasi yang akan menghapus semua karyawan dari tabel lesson7_employees yang juga ada di tabel job_history.

→
```
CREATE TABLE lesson7_emp

AS ( SELECT * FROM employees);


DELETE FROM lesson7_emp

WHERE employee_id IN ( SELECT DISTINCT employee_id
      FROM job_history);


SELECT * FROM lesson7_emp;
```

Result :

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	BONUS
100	Steven	King	SKING	515.123.4567	06/17/1987	AD_PRES	24000	-	-	90	-
205	Shelley	Higgins	SHIGGINS	515.123.8080	06/07/1994	AC_MGR	12000	-	101	110	-
206	William	Gietz	WGIETZ	515.123.8181	06/07/1994	AC_ACCOUNT	8300	-	205	110	-
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	01/29/2000	SA_MAN	10500	.2	100	80	1500
174	Ellen	Abel	EABEL	011.44.1644.429267	05/11/1996	SA_REP	11000	.3	149	80	1700
178	Kimberely	Grant	KGRANT	011.44.1644.429263	05/24/1999	SA_REP	7000	.15	149	-	-
124	Kevin	Mourgos	KMOURGOS	650.123.5234	11/16/1999	ST_MAN	5800	-	100	50	-
141	Trenna	Rajs	TRAJS	650.121.8009	10/17/1995	ST_CLERK	3500	-	124	50	-
142	Curtis	Davies	CDAVIES	650.121.2994	01/29/1997	ST_CLERK	3100	-	124	50	-
143	Randall	Matos	RMATOS	650.121.2874	03/15/1998	ST_CLERK	2600	-	124	50	-

9. Jalankan 3 pernyataan berikut untuk membuat 3 tabel baru untuk digunakan dalam pernyataan penyisipan Multi-tabel. Semua 3 tabel harus kosong pada saat pembuatan, karenanya kondisi WHERE 1=2 dalam klausa WHERE.

```
CREATE TABLE sal_history (employee_id, hire_date,
salary) AS SELECT employee_id, hire_date, salary
FROM employees
WHERE 1=2;

CREATE TABLE mgr_history (employee_id, manager_id,
salary) AS SELECT employee_id, manager_id, salary
FROM employees
WHERE 1=2;

CREATE TABLE special_sal (employee_id, salary)
AS SELECT employee_id, salary
FROM employees WHERE 1=2;
```

Setelah tabel ada di akun Anda, tulis pernyataan penyisipan Multi-Tabel untuk memilih terlebih dahulu employee_id, hire_date, salary, dan manager_id dari semua karyawan. Jika gaji lebih dari 20000 masukkan employee_id dan salary ke dalam tabel special_sal. Masukkan detail employee_id, hire_date, dan salary ke dalam tabel

sal_history. Masukkan employee_id, manager_id, dan salary ke dalam tabel mgr_history.

Anda harus mendapatkan pesan kembali yang mengatakan 39 baris telah dimasukkan. Verifikasi Anda mendapatkan pesan ini dan verifikasi Anda memiliki jumlah baris berikut di setiap tabel:

Sal_history: 19 baris

Mgr_history: 19 baris

Spesial_sal: 1

→
```
CREATE TABLE sal_history (employee_id, hire_date,
salary) AS SELECT employee_id, hire_date, salary
FROM employees
```

```
WHERE 1=2;

CREATE TABLE mgr_history (employee_id, manager_id,
     salary) AS SELECT employee_id, manager_id,
     salary
FROM employees
WHERE 1=2;

CREATE TABLE special_sal (employee_id, salary)
AS SELECT employee_id, salary
FROM employees WHERE 1=2;

INSERT FIRST
  WHEN salary > 20000 THEN
    INTO special_sal (employee_id, salary)
     VALUES(employee_id, salary)
  WHEN salary <= 20000 THEN
    INTO sal_history (employee_id, hire_date,
     salary) VALUES(employee_id, hire_date,
     salary)
  INTO mgr_history (employee_id, manager_id,
     salary) VALUES(employee_id, manager_id,
     salary)
SELECT employee_id, salary, hire_date, manager_id
     FROM employees;

SELECT COUNT(*) FROM sal_history;
```

```
SELECT COUNT(*) FROM mgr_history;


SELECT COUNT(*) FROM special_sal;
```

Result :

Sal_history:

COUNT(*)
19

Mgr_history:

COUNT(*)
19

Spesial_sal:

COUNT(*)
1