



SEPTEMBER 13, 2024

# LAB 2 - DATABASE DEVELOPMENT PROCESS

IS 6420-001 FALL 2024 DATABASE THEORY/DESIGN



ALI ABBAS LADHA  
U1537479

# Lab 2 SQL

## Guided Exercises

### Warranty Table

DBeaver 24.1.5 - <postgres> Script-6

```
postgres => DROP TABLE IF EXISTS warranty;
postgres => CREATE TABLE warranty (
postgres =>   policy_id VARCHAR(20) NOT NULL,
postgres =>   policy_name VARCHAR(50),
postgres =>   term_months INTEGER,
postgres =>   PRIMARY KEY (policy_id)
postgres => );
postgres => SELECT *
postgres => FROM warranty;
```

warranty 1

Grid	AZ policy_id	AZ policy_name	0: term_months
1	p01	3-year-warranty	36

Text

Record

Refresh Save Cancel Export data 1 1 row(s) fetched - 0.001s, on 2024-09-13 at 18:11:17

Output

```
Enter a part of a message
table "warranty" does
table "workers" does
table "computer" does
table "assembly" does
```

## Worker Table

DBeaver 24.1.5 - <postgres> Script-6

postgres public@is6420lab2

postgres \*<postgres> Script-6

```
26
27
28
29
30
31
32 DROP TABLE IF EXISTS workers;
33
34
35 CREATE TABLE worker (
36     employee_id VARCHAR(20) NOT NULL,
37     first_name VARCHAR(50),
38     last_name VARCHAR(50),
39     DOB DATE,
40     PRIMARY KEY (employee_id)
41 )
42
43
44
45
46
47
48
49
50
51
52
53
54 SELECT * FROM worker
55
```

Output

```
Enter a part of a mes
table "warranty" doe
table "workers" doe
table "computer" doe
table "assembly" doe
```

worker 1

Grid

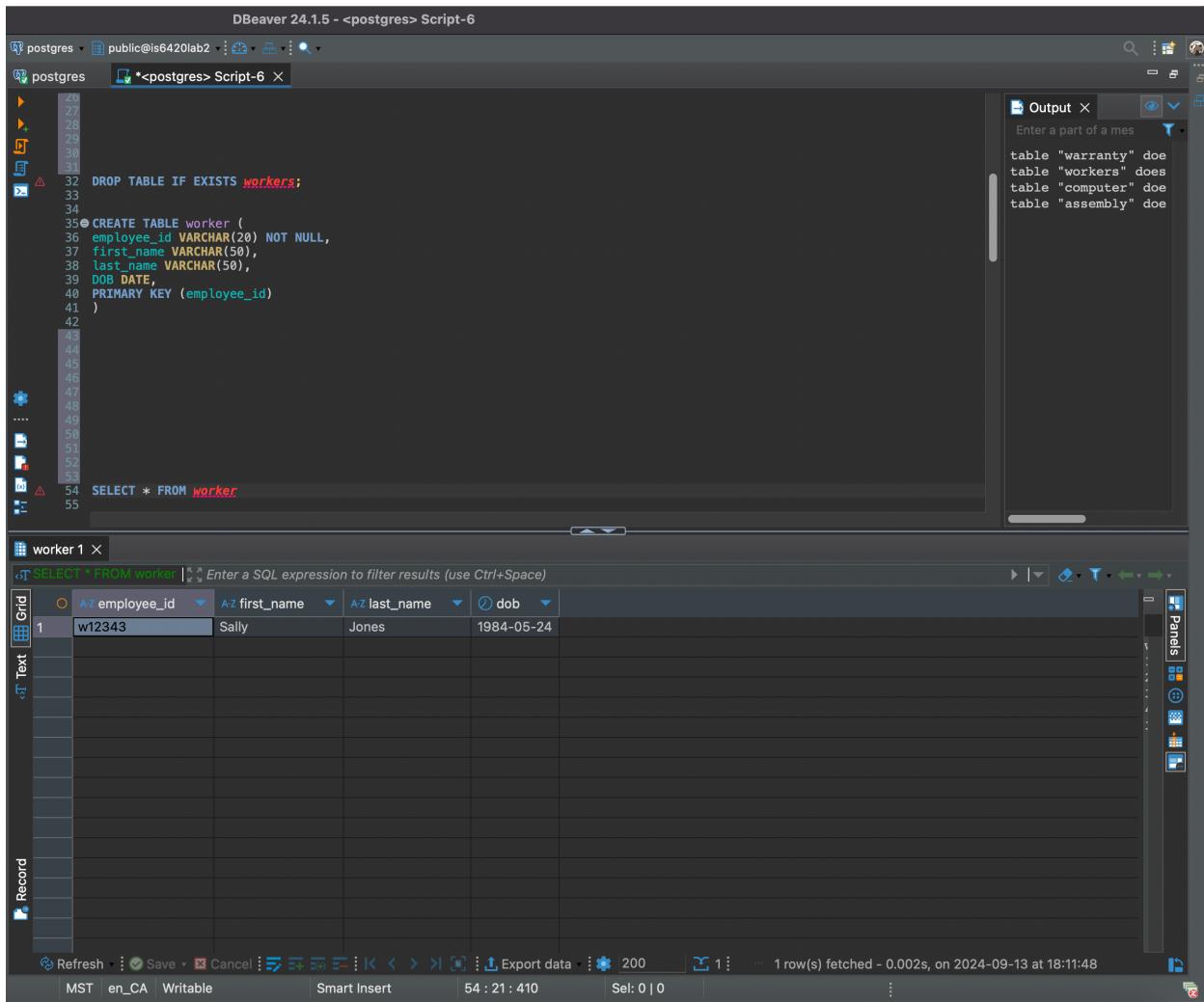
	AZ employee_id	AZ first_name	AZ last_name	dob
1	w12343	Sally	Jones	1984-05-24

Text

Record

Refresh Save Cancel Export data 200 1 1 row(s) fetched - 0.002s, on 2024-09-13 at 18:11:48

MST en\_CA Writable Smart Insert 54 : 21 : 410 Sel: 0 | 0



## Computer Table

DBeaver 24.1.5 - <postgres> Script-6

postgres public@ls6420lab2

\*<postgres> Script-6

```
56
57
58
59
60
61
62 DROP TABLE IF EXISTS computer;
63
64 CREATE TABLE computer(
65     serial_number VARCHAR(20) NOT NULL,
66     cpu_speed_ghz NUMERIC(6,3),
67     memory_size_gb NUMERIC(8,3),
68     price MONEY,
69     dvd_speed_rpm NUMERIC(8,3),
70     policy_id VARCHAR(20),
71     PRIMARY KEY(serial_number),
72     FOREIGN KEY(policy_id) REFERENCES warranty(policy_id)
73 )
74
75 SELECT * FROM computer
76
77
78
79
80
81
82
83
84
```

Output

Enter a part of a message

table "warranty" does  
table "workers" does  
table "computer" does  
table "assembly" does

computer 1

SELECT \* FROM computer

Grid	AZ serial_number	0..9 cpu_speed_ghz	0..9 memory_size_gb	0..9 price	0..9 dvd_speed_rpm	AZ policy_id
1	s1234	3.5	8	\$1,299.99	1,600	p01
Text						
Record						

Refresh Save Cancel Export data 200 1 1 row(s) fetched - 0.001s, on 2024-09-13 at 18:13:27

## Assembly Table

DBeaver 24.1.5 - <postgres> Script-6

postgres > <postgres> Script-6

```
86 DROP TABLE IF EXISTS assembly;
87
88 CREATE TABLE assembly (
89 employee_id VARCHAR(20),
90 serial_number VARCHAR(20),
91 total_hours NUMERIC(6,3),
92 PRIMARY KEY(employee_id, serial_number),
93 FOREIGN KEY (employee_id) REFERENCES worker (employee_id),
94 FOREIGN KEY (serial_number) REFERENCES computer (serial_number)
95 )
96
97 SELECT *
98 FROM assembly;
```

assembly 1 >

SELECT \* FROM assembly | Enter a SQL expression to filter results (use Ctrl+Space)

Grid	A-Z employee_id	A-Z serial_number	0-9 total_hours
1	w12343	s1234	4.75

Text

Record

Refresh Save Cancel Export data 200 1 1 row(s) fetched - 0.001s, on 2024-09-13 at 18:14:34

MST en\_CA Writable Smart Insert 107 : 1 : 1086 Sel: 0 | 0

Output

Enter a part of a message

table "warranty" does  
table "workers" does  
table "computer" does  
table "assembly" does

## Contact Table

DBeaver 24.1.5 - <postgres> Script-6

postgres \*<postgres> Script-6

```
100
101
102
103
104
105
106
107
108
109
110 CREATE TABLE contact(
111     employee_id VARCHAR(20) NOT NULL,
112     phone_number VARCHAR(10) NOT NULL,
113     PRIMARY KEY (employee_id, phone_number),
114     FOREIGN KEY (employee_id) REFERENCES worker (employee_id)
115 )
116
117
118 SELECT *
119 FROM contact;
```

contact 1

Grid

	employee_id	phone_number
1	w12343	8015551111

Text

Record

Refresh Save Cancel Export data 200 1 1 row(s) fetched - 0.001s, on 2024-09-13 at 18:15:06

Output

```
table "warranty" doe
table "workers" does
table "computer" doe
table "assembly" doe
```

# Challenge Exercises – Part 1

Add more data on Tables:

## Warranty Table

The screenshot shows a database grid interface with the title "warranty 1". The query bar at the top displays "SELECT \* FROM warranty ORDER BY policy". A tooltip "Enter a SQL expression to filter results (use Ctrl+Space)" is visible. The grid has columns: Grid, Text, and Record. The Record column contains icons for Refresh, Save, Cancel, and various navigation and export functions. The data in the grid is as follows:

	Grid	AZ policy_id	AZ policy_name	0.9 term_months
1		p01	3-year-warranty	36
2		p02	4-year-warranty	48
3		p03	5-year-warranty	60
4		p04	6-year-warranty	66
5		p05	7-year-warranty	76

## Worker Table

The screenshot shows a database grid interface with the title "worker 1". The query bar at the top displays "SELECT \* FROM worker". A tooltip "Enter a SQL expression to filter results (use Ctrl+Space)" is visible. The grid has columns: Grid, Text, and Record. The Record column contains icons for Refresh, Save, Cancel, and various navigation and export functions. The data in the grid is as follows:

	Grid	AZ employee_id	AZ first_name	AZ last_name	0.9 dob
1		w12343	Sally	Jones	1984-05-24
2		w12344	John	Green	1985-01-01
3		w12345	Sarah	Green	1989-02-05
4		w12346	Tyler	Blue	1987-01-17
5		w12347	Kyle	Blue	1990-04-02

## Computer Table

A screenshot of a database grid titled "computer 1". The grid displays a table with the following columns: serial\_number, cpu\_speed\_ghz, memory\_size\_gb, price, dvd\_speed\_rpm, and policy\_id. The data consists of five rows:

Grid	serial_number	cpu_speed_ghz	memory_size_gb	price	dvd_speed_rpm	policy_id
1	s1234	3.5	8	\$1,299.99	1,600	p01
2	s12345	4	9	\$1,399.99	1,700	p02
3	s12346	5	10	\$1,499.99	1,800	p03
4	s12347	5	10	\$1,499.99	1,800	p04
5	s12348	6	11	\$1,599.99	2,100	p05

Record

Refresh Save Cancel Export data 200 5 5 row(s) fetched - 0.001s, on 2024-09-13 at 18:31:53

## Assembly Table

A screenshot of a database grid titled "assembly 1". The grid displays a table with the following columns: employee\_id, serial\_number, and total\_hours. The data consists of four rows:

Grid	employee_id	serial_number	total_hours
1	w12343	s1234	4.75
2	w12344	s1234	5
3	w12345	s12347	7
4	w12347	s12348	8

Record

Refresh Save Cancel Export data 200 4 4 row(s) fetched - 0.001s, on 2024-09-13 at 18:35:47

## Contact table

A screenshot of a database grid titled "contact 1". The grid displays a table with two columns: "employee\_id" and "phone\_number". The data consists of five rows:

Grid	employee_id	phone_number
1	w12343	8015551111
2	w12344	123456789
3	w12345	543234091
4	w12346	908123456
5	w12347	890987612

The interface includes a toolbar with various icons for filtering, sorting, and exporting data. A status bar at the bottom indicates "5 row(s) fetched - 0.002s, on 2024-09-13 at 18:38:03".

Practice on Null:

2a)

A screenshot of a database grid titled "warranty 1". The grid displays a table with three columns: "policy\_id", "customer\_id", and "warranty\_pkey". The data consists of two rows:

policy_id	customer_id	warranty_pkey
p02	null	2

An error message is displayed on the left side of the screen:

SQL Error [23505]: ERROR: duplicate key value violates unique constraint "warranty\_pkey"  
Detail: Key (policy\_id)=(p02) already exists.

The error message also appears in the status bar at the bottom: "SQL Error [23505]: ERROR: duplicate key value violates unique constraint \"warranty\_pkey\" Detail: Key (pol".

2b i)

warranty 1 X

```
SELECT * FROM warranty | Enter a SQL expression to filter results (use Ctrl+Space)
```

**SQL Error [23502]: ERROR: null value in column "policy\_id" of relation "warranty" violates not-null constraint**  
 Detail: Failing row contains (null, 5 year warranty, 5).

Error position:

```
1 INSERT INTO warranty
2   VALUES (null, '5 year warranty', 5)
```

Details >>      

Refresh Save Cancel Export data 200 5

**SQL Error [23502]: ERROR: null value in column "policy\_id" of relation "warranty" violates not-null constraint**

**2b ii)** Does not work since we have added a NOT NULL clause to the policy\_id field for the Warranty table

**3a i)**

computer 1 X

```
SELECT * FROM comput | Enter a SQL expression to filter results (use Ctrl+Space)
```

**SQL Error [23503]: ERROR: update or delete on table "warranty" violates foreign key constraint "computer\_policy\_id\_fkey" on table "computer"**  
 Detail: Key (policy\_id)=(p01) is still referenced from table "computer".

Error position:

```
1 DELETE FROM warranty
2 WHERE policy_id = 'p01'
```

Details >>      

Refresh Save Cancel Export data 200 5

**SQL Error [23503]: ERROR: update or delete on table "warranty" violates foreign key constraint "computer\_pc**

### 3a ii)

Does Not work since we linked a foreign key to the computer table thus maintaining referential integrity. Therefore, it cannot be deleted since the computer table has the value of p01 too in its rows.

## Challenge Exercise – Part 2

### 2. Create a query that displays all of the employees who worked on a computer

The screenshot shows a database interface with a code editor at the top containing a SQL query:

```
243  
244  
245 SELECT  
246     c.serial_number AS computer_serial_number,  
247     w.first_name AS employee_first_name,  
248     w.last_name AS employee_last_name  
249 FROM computer AS c  
250 INNER JOIN assembly AS a  
251 USING(serial_number)  
252 INNER JOIN worker AS w  
253 ON a.employee_id = w.employee_id  
254 ORDER BY c.serial_number ASC, w.last_name ASC;  
255  
256
```

Below the code editor is a grid view showing the results of the query:

Grid	computer_serial_number	employee_first_name	employee_last_name
1	s1234	John	Green
2	s1234	Sally	Jones
3	s12347	Sarah	Green
4	s12348	Kyle	Blue

At the bottom of the interface, there are various buttons for refresh, save, cancel, and export, along with a status message: "4 row(s) fetched - 0.002s, on 2024-09-13 at 18:52:28".

3. Create a query that displays the total number of labor hours spent on each computer

The screenshot shows a database interface with a code editor at the top and a results grid below. The code editor contains the following SQL query:

```
271
272     SELECT
273         SUM(a.total_hours) AS labor_hours,
274         a.serial_number AS computer_serial_number
275     FROM assembly AS a
276     GROUP BY computer_serial_number
277     ORDER BY labor_hours DESC
278
279
280
281
282
```

The results grid, titled "assembly 1", displays the following data:

	labor_hours	computer_serial_number
1	9.75	s1234
2	8	s12348
3	7	s12347

At the bottom of the interface, there are various navigation and export buttons, along with a status message indicating "3 row(s) fetched - 0.002s, on 2024-09-13 at 18:56:50".