Final Group Project

CIS 215
Prepared By: Alexandre Paquette
Partner(s): Marc-Andre Gingras
Prepared For: Kamaljeet Batra

Canadore College

December 09, 2021

Index

Problem Description

Data Dictionary

ER Diagram

User Diagram

Queries

Problem Description

Our project aims to simplify and streamline the day-to-day operations of a small computer repair shop. The system is intended to be used by the staff only, and will be used by employees with several different roles. When a customer brings in or sends the shop their problematic devices, a customer service representative (CSR) will begin the intake process. Customer information is collected, as well as information about the computer such as model number, name, brand, and issue(s) as described by the customer. The CSR then creates the repair order. From there. technicians, who are responsible for the actual repair work, will then be tasked with working on the new repair order, updating the repair status as the job progresses. Technicians are able to access inventory tables which contain information about all of the parts the store currently has in stock. Technicians are also capable of producing part requests, in which the technician will be requesting stock out of inventory for the repair. The inventory clerks are then responsible with completing these part requests for the technician, as well as generally maintaining the inventory system. From an administrative perspective, the employee table provides information to managers about each employee, including their department id, salary, and their direct manager. This system allows the computer repair shop to simplify the intake process, as well as organize and structure the key elements of the business in such a way that a good workflow can occur. The overall purpose, in essence, is to reduce employee system usage, which frees up CSRs time to better interact with customers, Technicians to spend more time working on repairs, and aids managers in understanding the current state of the business.

Data Dictionary

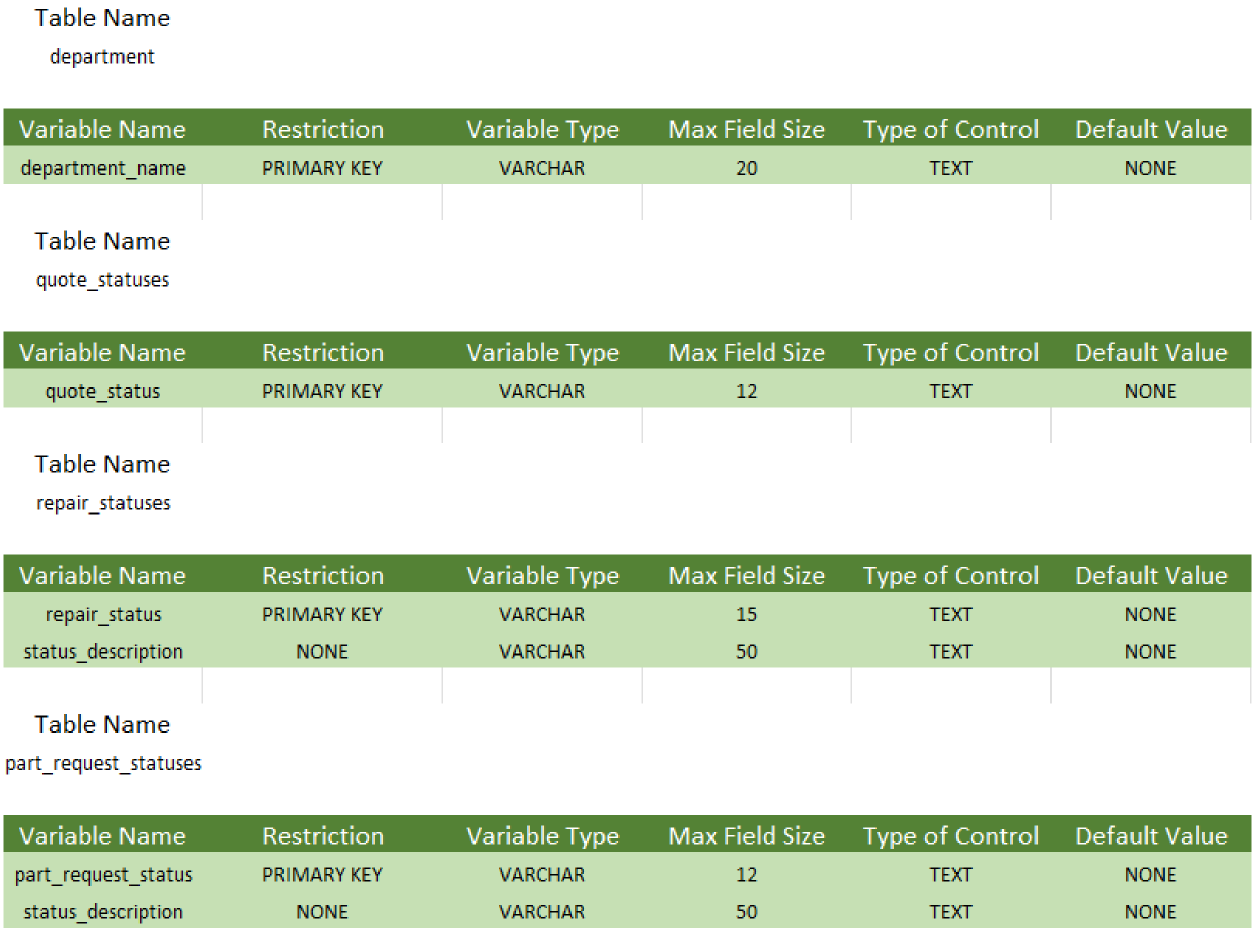


Table Name

department

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
department_name	PRIMARY KEY	VARCHAR	20	TEXT	NONE

Table Name

quote_statuses

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
quote_status	PRIMARY KEY	VARCHAR	12	TEXT	NONE

Table Name

repair_statuses

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
repair_status	PRIMARY KEY	VARCHAR	15	TEXT	NONE
status_description	NONE	VARCHAR	50	TEXT	NONE

Table Name

part_request_statuses

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
part_request_status	PRIMARY KEY	VARCHAR	12	TEXT	NONE
status_description	NONE	VARCHAR	50	TEXT	NONE

Table Name

computer_models

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
model_id	PRIMARY KEY	VARCHAR	20	TEXT	NONE
product_name	NONE	VARCHAR	50	TEXT	NONE
brand	NONE	VARCHAR	30	TEXT	NONE

Table Name

employees

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
employee_id	PRIMARY KEY	INT	4	TEXT	NONE
first_name	NONE	VARCHAR	20	TEXT	NONE
last_name	NONE	VARCHAR	20	TEXT	NONE
salary	NOT NULL	DOUBLE	8	TEXT	NONE
manager	FOREIGN KEY (employees)	INT	4	SELECT OPTION	NONE
department	FOREIGN KEY (departments), NOT NULL	VARCHAR	20	SELECT OPTION	NONE

Table Name

customer_contacts

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
customer_id	PRIMARY KEY	INT	4	TEXT	NONE
first_name	NONE	VARCHAR	20	TEXT	NONE
last_name	NONE	VARCHAR	20	TEXT	NONE
email	NONE	VARCHAR	50	TEXT	NONE
phone_number	NONE	VARCHAR	15	TEXT	NONE

Table Name

repairs

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
service_id	PRIMARY KEY	INT	4	TEXT	NONE
technician_id	FOREIGN KEY (employees)	INT	4	TEXT	NONE
model_id	FOREIGN KEY (computer_models)	VARCHAR	20	TEXT	NONE
repair_summary	NONE	VARCHAR	300	TEXT	NONE
serial_number	NOT NULL	VARCHAR	50	TEXT	NONE
received_date	NOT NULL	DATE	10	TEXT	NONE
completed_date	NONE	DATE	10	TEXT	NONE
repair_status	FOREIGN KEY (repair_statuses), NOT NULL	VARCHAR	15	SELECT OPTION	"received"

[redacted]

Table Name

billable_quotes

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
quote_id	PRIMARY KEY FOREIGN KEY (repairs)	INT	4	TEXT	NONE
csr_id	FOREIGN KEY (employees)	INT	4	TEXT	NONE
customer_id	FOREIGN KEY (customer_contacts)	INT	4	TEXT	NONE
total	NONE	DOUBLE	7	TEXT	NONE
quote_status	FOREIGN KEY (quote_statuses), NOT NULL	VARCHAR	12	SELECT OPTION	"pending"
date_sent	NONE	DATE	10	TEXT	NONE
date_paid	NONE	DATE	10	TEXT	NONE

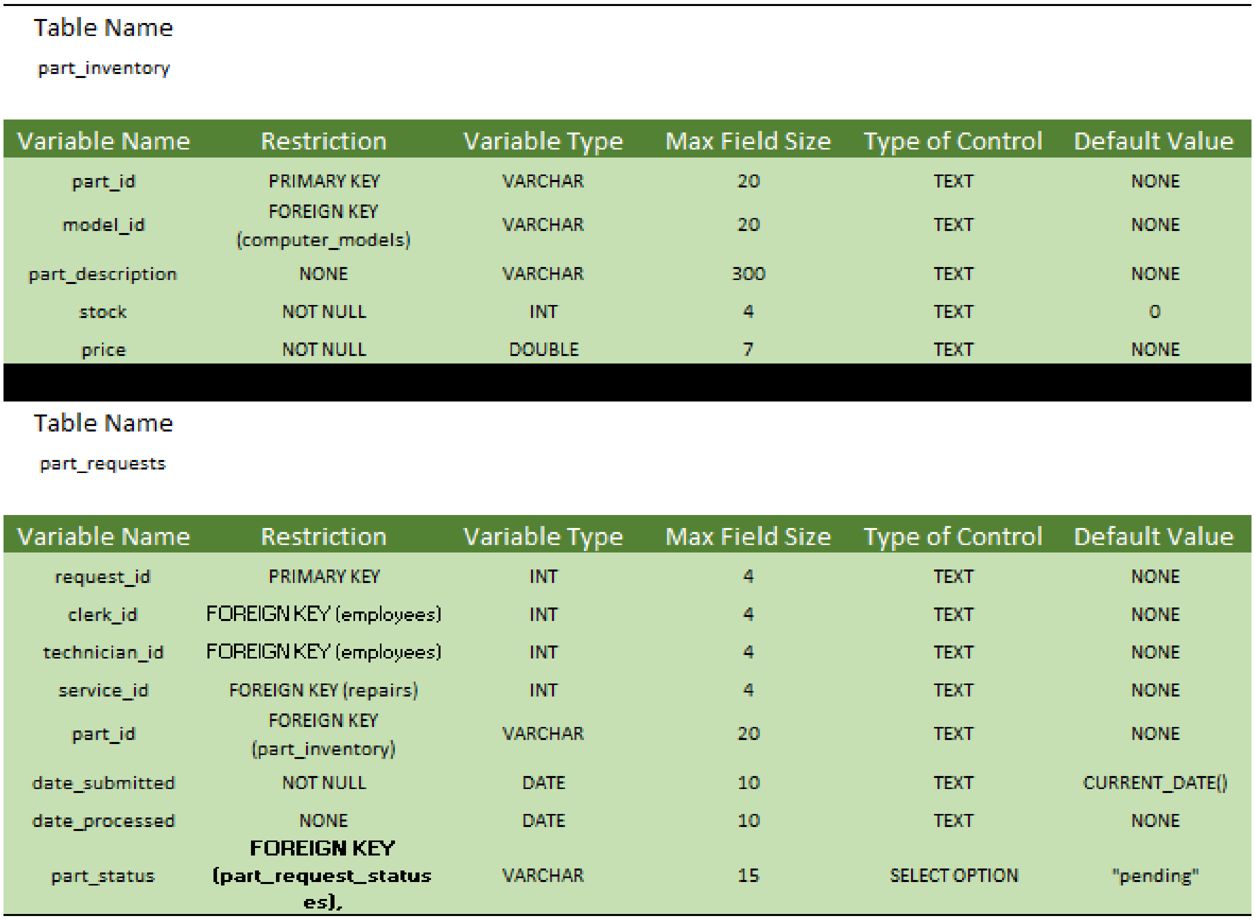


Table Name

part_inventory

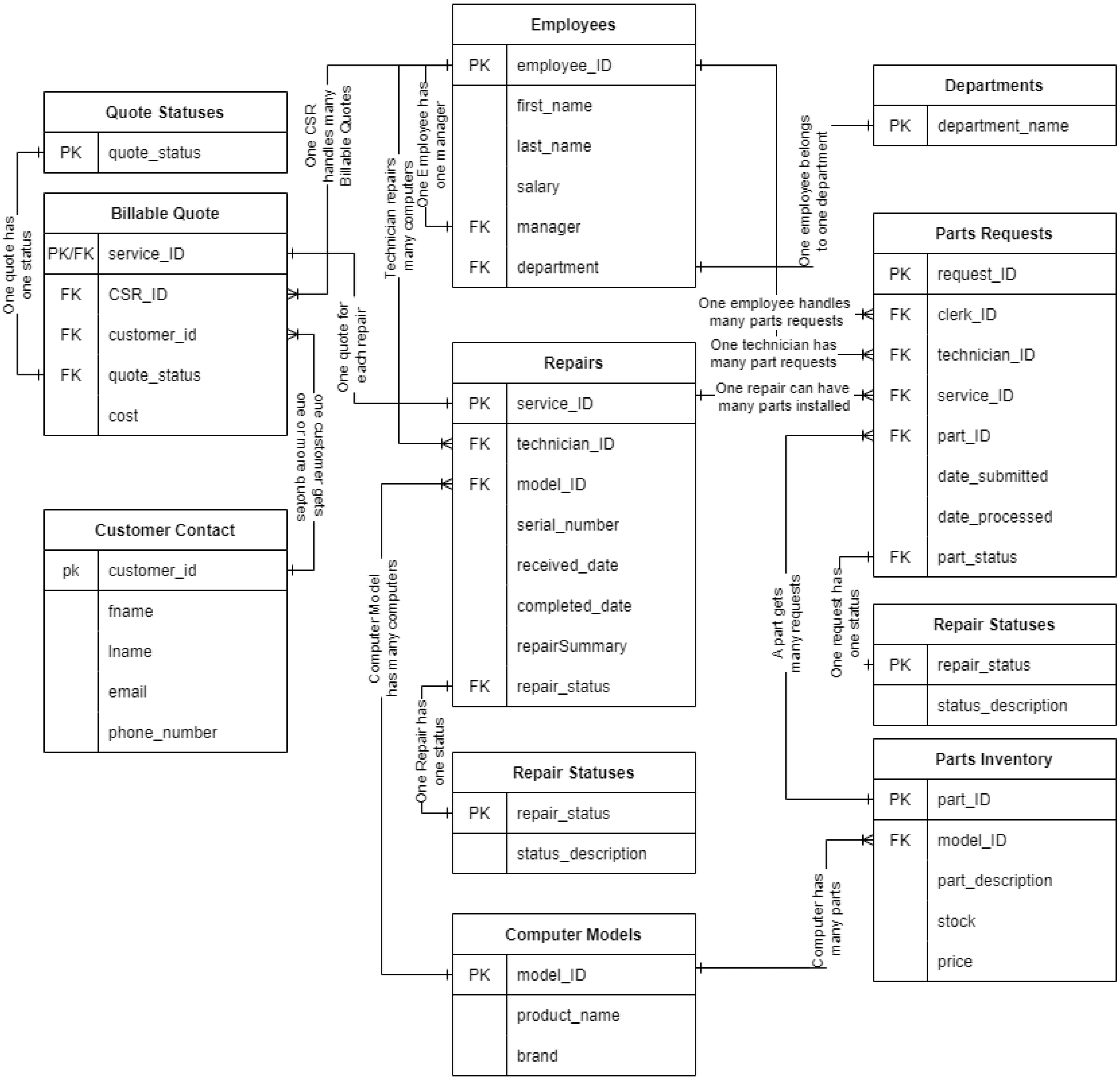
Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
part_id	PRIMARY KEY	VARCHAR	20	TEXT	NONE
model_id	FOREIGN KEY (computer_models)	VARCHAR	20	TEXT	NONE
part_description	NONE	VARCHAR	300	TEXT	NONE
stock	NOT NULL	INT	4	TEXT	0
price	NOT NULL	DOUBLE	7	TEXT	NONE

Table Name

part_requests

Variable Name	Restriction	Variable Type	Max Field Size	Type of Control	Default Value
request_id	PRIMARY KEY	INT	4	TEXT	NONE
clerk_id	FOREIGN KEY (employees)	INT	4	TEXT	NONE
technician_id	FOREIGN KEY (employees)	INT	4	TEXT	NONE
service_id	FOREIGN KEY (repairs)	INT	4	TEXT	NONE
part_id	FOREIGN KEY (part_inventory)	VARCHAR	20	TEXT	NONE
date_submitted	NOT NULL	DATE	10	TEXT	CURRENT_DATE()
date_processed	NONE	DATE	10	TEXT	NONE
part_status	**FOREIGN KEY (part_request_statuses),**	VARCHAR	15	SELECT OPTION	"pending"

Entity relationship diagram



User Information

Administrator

Administrators manage the repair center. They can manage all aspects of the database. Administrators will have read/write access to all tables in the database

Managers

Managers are responsible for managing an aspect of the business (inventory, computer repairs, or customer service). In addition to all the permissions of their subordinates, managers also have access to the employee table, but only for employees they manage. Table views will be created for each type of manager.

Warehouse Clerk

Inventory clerks are responsible for processing parts requests from the technicians. They will have read/write access to the parts requests and parts inventory tables.

Table	Read/Write	Reason for access
Parts Inventory	Read/Write	Inventory clerks need access to the parts inventory so that they can update stock and price of parts as they have consumed or purchased
Parts Requests	Read/Write	Inventory clerks need access to the parts requests so they can process requests from technicians

Computer Technician

Computer technicians are responsible for performing repairs on computers.

Table	**Read/Write**	**Reason for access**
Repairs	Read/Write	Technicians need access to the repairs table to update repair status and write a repair summary for records
Parts Inventory	Read	Technicians need to view the part list for the computers they repair and so they can determine what parts are needed for repairs.
Parts Requests	Write	Technicians need to be able to submit parts requests to the inventory clerks. They do not need to view these requests

Customer Service Representative

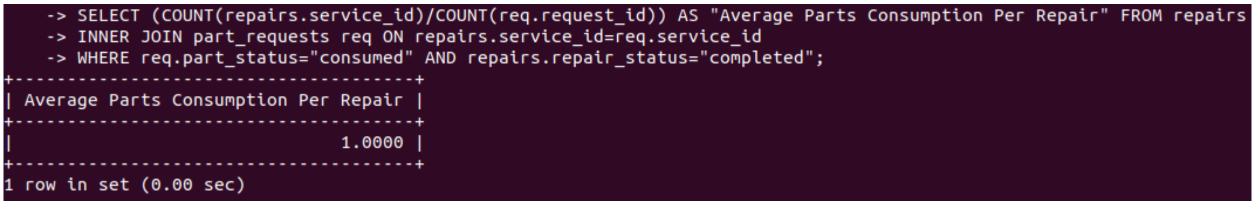
CSRs are responsible for interacting with customers. They will take the information from the customer for the service order, and they will also process payment from the customer for repairs.

Table	**Read/Write**	**Reason for access**
Billable quote	Read/Write	CSRs will need access to create, view, and modify billable quotes
Customer Contact	Read/Write	CSRs will need access to create, view, and modify customer contact information
Computer Models	Write	CSRs will need access to create a record of each computer model. They only need write permission as they will not use this information for any other tasks. This is only necessary for models that have never been in for repair before
Parts Inventory	Read/Write	CSRs will need access to create records for parts for models that are in for repairs. They will need to reference this table when creating quotes to determine the price of parts needed for repair
Parts Requests	Read	CSRs need to view parts requested by technicians to determine how much to charge customers
Repairs	Read	CSRs need to view repair details to determine billable quote

Queries

This query will calculate the average parts consumption per completed repair. Parts are filtered to include those that are consumed and repairs are filtered to include those that are completed.

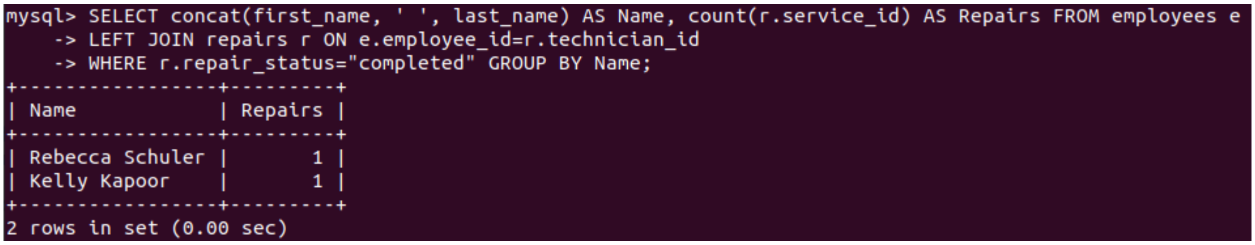
```
SELECT (COUNT(repairs.service_id)/COUNT(req.request_id)) AS "Average Parts Consumption Per Repair"
FROM repairs

        INNER JOIN part_requests req ON repairs.service_id=req.service_id

                WHERE req.part_status="consumed" AND repairs.repair_status="completed";
```



```
    -> SELECT (COUNT(repairs.service_id)/COUNT(req.request_id)) AS "Average Parts Consumption Per Repair" FROM repairs
    -> INNER JOIN part_requests req ON repairs.service_id=req.service_id
    -> WHERE req.part_status="consumed" AND repairs.repair_status="completed";
+--------------------------------------+
| Average Parts Consumption Per Repair |
+--------------------------------------+
|                               1.0000 |
+--------------------------------------+
1 row in set (0.00 sec)
```

This query will display how many repairs each technician has completed. The repairs are filtered to only include those that have been completed, and the queries are grouped by technician. The repairs and employee tables are joined by technician id.

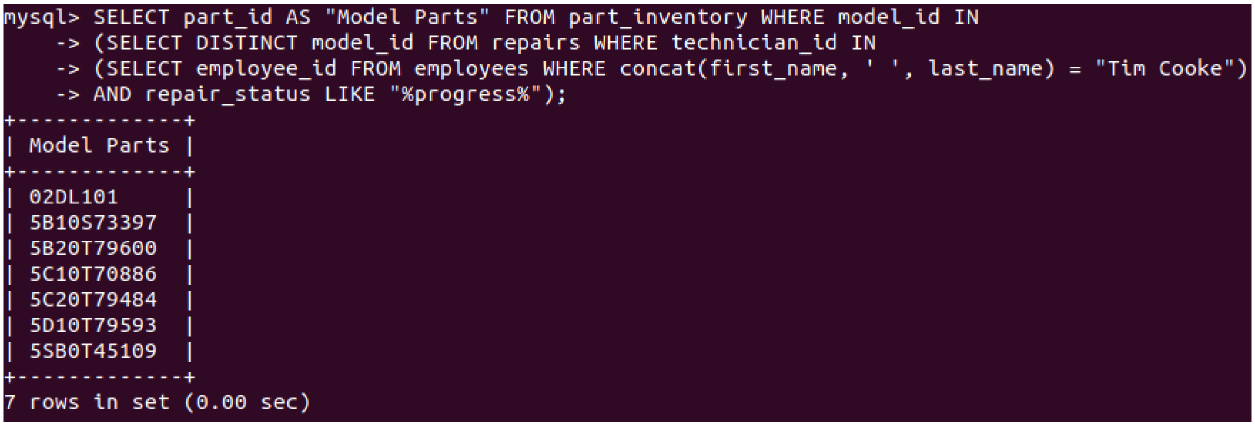
```
SELECT concat(first_name, ' ', last_name) AS Name, count(r.service_id) AS Repairs from employees e

        LEFT JOIN repairs r ON e.employee_id=r.technician_id

                WHERE r.repair_status="completed" GROUP BY Name;
```



```
mysql> SELECT concat(first_name, ' ', last_name) AS Name, count(r.service_id) AS Repairs FROM employees e
    -> LEFT JOIN repairs r ON e.employee_id=r.technician_id
    -> WHERE r.repair_status="completed" GROUP BY Name;
+-----------------+---------+
| Name            | Repairs |
+-----------------+---------+
| Rebecca Schuler |       1 |
| Kelly Kapoor    |       1 |
+-----------------+---------+
2 rows in set (0.00 sec)
```

This query will list all the parts for models which is in repair with a specific technician and is in progress. The repair status is filtered to include the keyword “progress” (short for in progress) and the technician is specified by concatenating first and last name for t he comparison. The distinct model ids the technician is working on are returned and corresponding parts are listed.

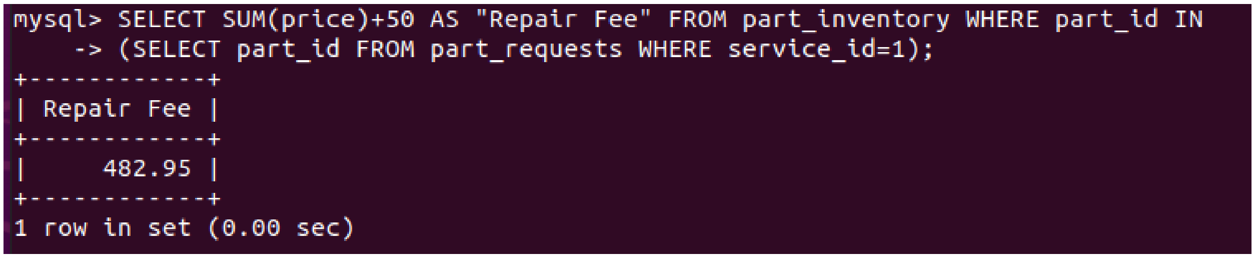
```
SELECT part_id AS "Model Parts" FROM part_inventory WHERE model_id IN

        (SELECT DISTINCT model_id FROM repairs WHERE technician_id IN

                (SELECT employee_id FROM employees WHERE concat(first_name, ' ', last_name) =
                "Tim Cooke") AND repair_status LIKE "%progress%");
```



```
mysql> SELECT part_id AS "Model Parts" FROM part_inventory WHERE model_id IN
    -> (SELECT DISTINCT model_id FROM repairs WHERE technician_id IN
    -> (SELECT employee_id FROM employees WHERE concat(first_name, ' ', last_name) = "Tim Cooke")
    -> AND repair_status LIKE "%progress%");
+-------------+
| Model Parts |
+-------------+
| 02DL101     |
| 5B10S73397  |
| 5B20T79600  |
| 5C10T70886  |
| 5C20T79484  |
| 5D10T79593  |
| 5SB0T45109  |
+-------------+
7 rows in set (0.00 sec)
```

This query will calculate the price of a billable repair based on the sum of the price of parts requested for the repair. A service id is specified and all associated part requests are pulled. The sum of the price from the part inventory for each associated part request is calculated and the service fee is added to the total.

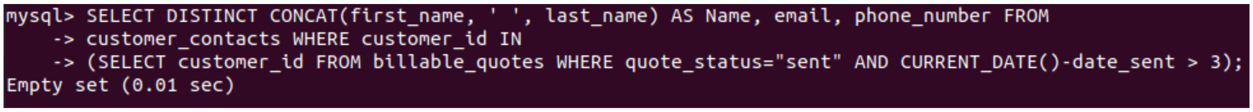
```
SELECT SUM(price)+50 AS "Repair Fee" FROM part_inventory WHERE part_id IN

        (SELECT part_id FROM part_requests WHERE service_id=1);
```



```
mysql> SELECT SUM(price)+50 AS "Repair Fee" FROM part_inventory WHERE part_id IN
    -> (SELECT part_id FROM part_requests WHERE service_id=1);
+------------+
| Repair Fee |
+------------+
|     482.95 |
+------------+
1 row in set (0.00 sec)
```

This query will display the customer information of customers who haven't paid their bills for more than three days. The subquery will return customer ids whose quotes have been sent and the current day minus the sate sent is greater than three. Distinct customers are displayed in case a single customer has multiple unpaid bills.

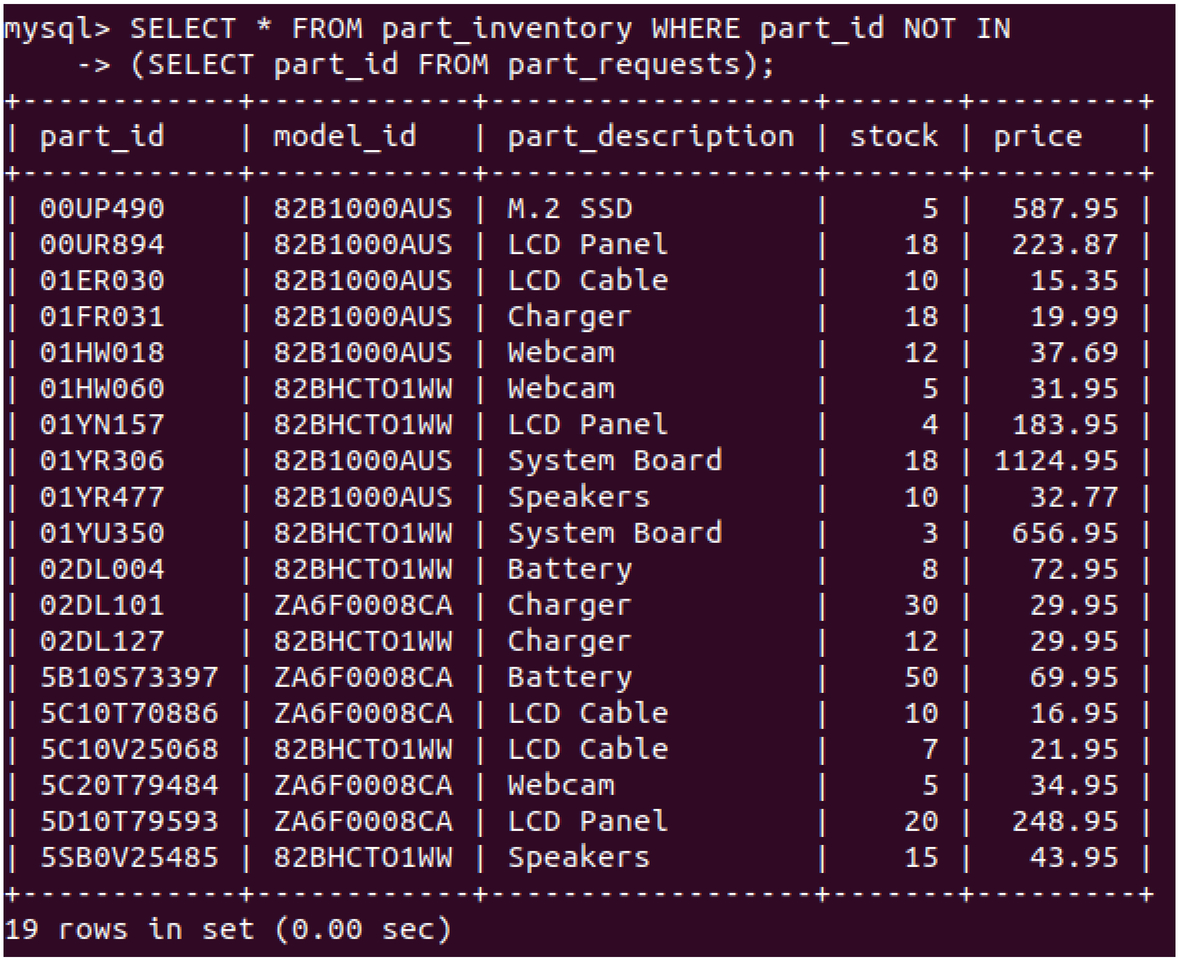
```
SELECT DISTINCT CONCAT(first_name, ' ', last_name) AS Name, email,
phone_number FROM customer_contacts WHERE customer_id IN

     (SELECT customer_id FROM billable_quotes WHERE
     quote_status="sent" AND CURRENT_DATE()-date_sent > 3);
```



```
mysql> SELECT DISTINCT CONCAT(first_name, ' ', last_name) AS Name, email, phone_number FROM
    -> customer_contacts WHERE customer_id IN
    -> (SELECT customer_id FROM billable_quotes WHERE quote_status="sent" AND CURRENT_DATE()-date_sent > 3);
Empty set (0.01 sec)
```

This query will display any part that was never requested. The subquery returns all part ids from part request, and the main query looks for parts in the inventory that are not included in that list.

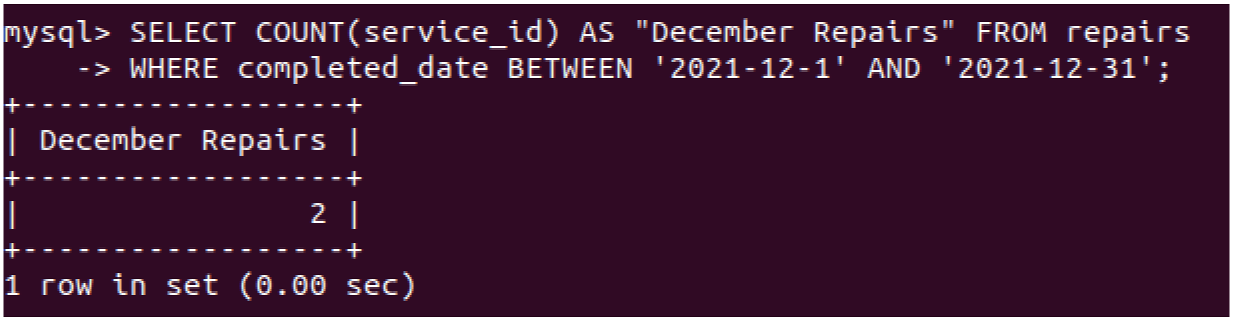
```
SELECT * FROM part_inventory WHERE part_id NOT IN

  (SELECT part_id FROM part_requests);
```



```
mysql> SELECT * FROM part_inventory WHERE part_id NOT IN
    -> (SELECT part_id FROM part_requests);
+------------+------------+------------------+-------+---------+
| part_id    | model_id   | part_description | stock | price   |
+------------+------------+------------------+-------+---------+
| 00UP490    | 82B1000AUS | M.2 SSD          |     5 |  587.95 |
| 00UR894    | 82B1000AUS | LCD Panel        |    18 |  223.87 |
| 01ER030    | 82B1000AUS | LCD Cable        |    10 |   15.35 |
| 01FR031    | 82B1000AUS | Charger          |    18 |   19.99 |
| 01HW018    | 82B1000AUS | Webcam           |    12 |   37.69 |
| 01HW060    | 82BHCTO1WW | Webcam           |     5 |   31.95 |
| 01YN157    | 82BHCTO1WW | LCD Panel        |     4 |  183.95 |
| 01YR306    | 82B1000AUS | System Board     |    18 | 1124.95 |
| 01YR477    | 82B1000AUS | Speakers         |    10 |   32.77 |
| 01YU350    | 82BHCTO1WW | System Board     |     3 |  656.95 |
| 02DL004    | 82BHCTO1WW | Battery          |     8 |   72.95 |
| 02DL101    | ZA6F0008CA | Charger          |    30 |   29.95 |
| 02DL127    | 82BHCTO1WW | Charger          |    12 |   29.95 |
| 5B10S73397 | ZA6F0008CA | Battery          |    50 |   69.95 |
| 5C10T70886 | ZA6F0008CA | LCD Cable        |    10 |   16.95 |
| 5C10V25068 | 82BHCTO1WW | LCD Cable        |     7 |   21.95 |
| 5C20T79484 | ZA6F0008CA | Webcam           |     5 |   34.95 |
| 5D10T79593 | ZA6F0008CA | LCD Panel        |    20 |  248.95 |
| 5SB0V25485 | 82BHCTO1WW | Speakers         |    15 |   43.95 |
+------------+------------+------------------+-------+---------+
19 rows in set (0.00 sec)
```

This query shows the number of repairs processed in December. A date range is provided for the completed date field and the service ids from the repairs table are counted.

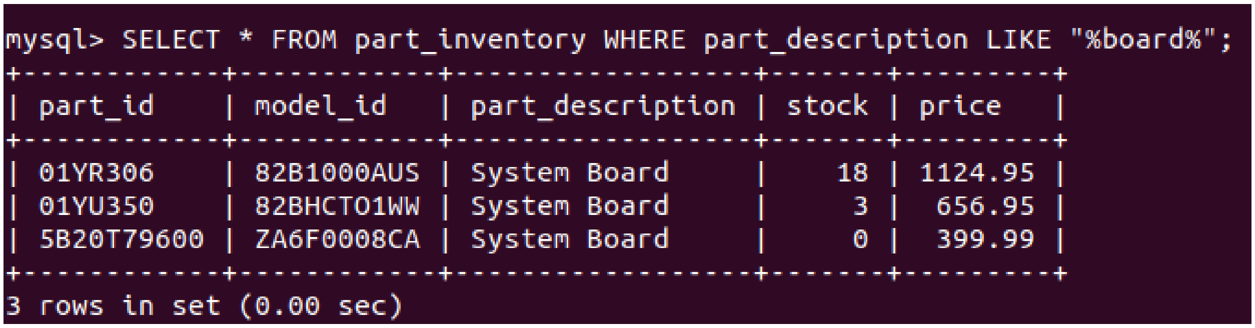
```
SELECT COUNT(service_id) AS "December Repairs" FROM repairs

     WHERE completed_date BETWEEN '2021-12-1' AND '2021-12-31';
```



```
mysql> SELECT COUNT(service_id) AS "December Repairs" FROM repairs
    -> WHERE completed_date BETWEEN '2021-12-1' AND '2021-12-31';
+------------------+
| December Repairs |
+------------------+
|                2 |
+------------------+
1 row in set (0.00 sec)
```

Returns parts that have the characters “board” in it. This is especially useful for technicians who just want to search for all parts with a keyword. This is easier to do than to search for a full term like “system board” or “sub board”.

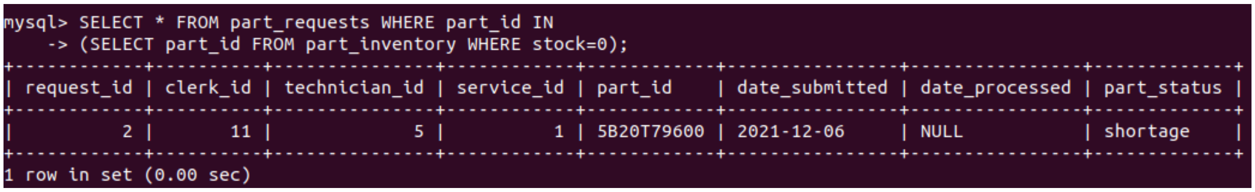
```
SELECT * FROM part_inventory WHERE part_description LIKE "%board%";
```



```
mysql> SELECT * FROM part_inventory WHERE part_description LIKE "%board%";
+------------+------------+------------------+-------+---------+
| part_id    | model_id   | part_description | stock | price   |
+------------+------------+------------------+-------+---------+
| 01YR306    | 82B1000AUS | System Board     |    18 | 1124.95 |
| 01YU350    | 82BHCTO1WW | System Board     |     3 |  656.95 |
| 5B20T79600 | ZA6F0008CA | System Board     |     0 |  399.99 |
+------------+------------+------------------+-------+---------+
3 rows in set (0.00 sec)
```

Shows parts that are requested but not in stock. The subquery returns all part ids where the stock is zero. The main query lists all parts request of parts that are in the list.

```
SELECT * FROM part_requests WHERE part_id IN

     (SELECT part_id FROM part_inventory WHERE stock=0);
```



```
mysql> SELECT * FROM part_requests WHERE part_id IN
    -> (SELECT part_id FROM part_inventory WHERE stock=0);
+------------+----------+---------------+------------+------------+----------------+----------------+-------------+
| request_id | clerk_id | technician_id | service_id | part_id    | date_submitted | date_processed | part_status |
+------------+----------+---------------+------------+------------+----------------+----------------+-------------+
|          2 |       11 |             5 |          1 | 5B20T79600 | 2021-12-06     | NULL           | shortage    |
+------------+----------+---------------+------------+------------+----------------+----------------+-------------+
1 row in set (0.00 sec)
```