



JOHN MCQUAIDE, JESSIE WILKINS, STEVEN OLSEN, YURI KHECHOYAN



Background: Why Qu · eu · ed exists?

► *Benefits:*

- Improve efficiency (B2B) with aiding customers
- Let CSRs & customers know who is next in line for assistance
- Give customers a more interactive experience
- Allow Business to Significantly Cut Costs
- No maintenance
- No stolen equipment



Putting things in Perspective

| Qu · eu · ed | Cost | Traditional Queue System | Cost |
|---|----------|--|--------------------------------|
|  | \$0.0075 |  | ~ \$100/pager |
| <ul style="list-style-type: none"> • Maintenance • Stolen | N/A |  | ~ \$500-\$1000 /transmitter |
| -- | -- | <ul style="list-style-type: none"> • Maintenance/Fix • Additional Pagers • Stolen | \$\$\$\$ |

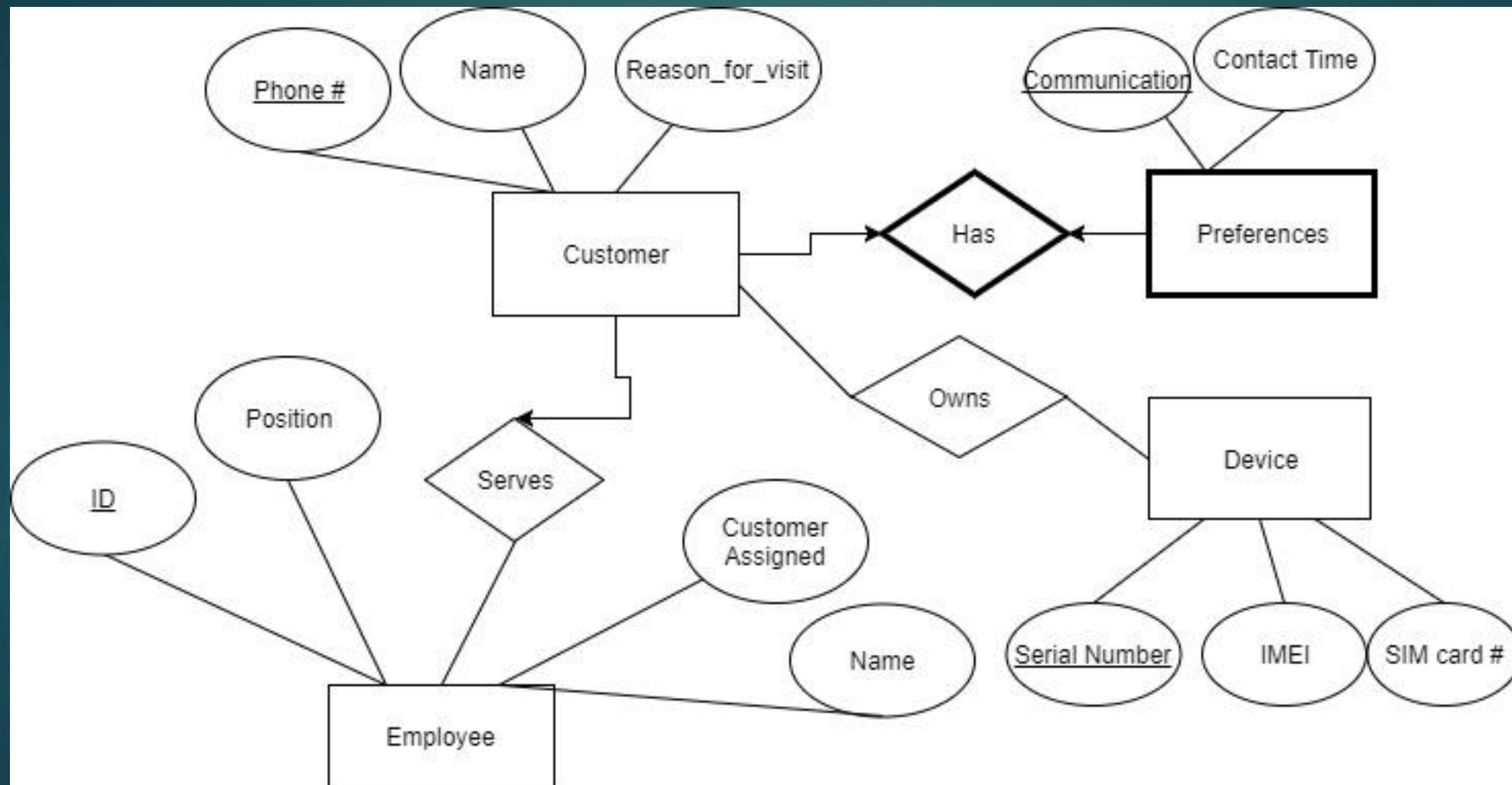


Table Implementation

| Table | Action | Rows | Type | Collation | Size | Overhead |
|-----------------------------------|---|------|--------|-------------------|--------|----------|
| <input type="checkbox"/> customer | ★ Browse Structure Search Insert Empty Drop | 3 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| <input type="checkbox"/> device | ★ Browse Structure Search Insert Empty Drop | 0 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| <input type="checkbox"/> employee | ★ Browse Structure Search Insert Empty Drop | 2 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| 3 tables | Sum | 5 | InnoDB | latin1_swedish_ci | 80 KiB | 0 B |

| # | Name | Type |
|----------------------------|----------------|-------------|
| <input type="checkbox"/> 1 | Fname | varchar(50) |
| <input type="checkbox"/> 2 | Lname | varchar(50) |
| <input type="checkbox"/> 3 | CellNum 🔑 | bigint(10) |
| <input type="checkbox"/> 4 | Email | varchar(75) |
| <input type="checkbox"/> 5 | ReasonForVisit | varchar(15) |
| <input type="checkbox"/> 6 | TicketID 🔑 | varchar(14) |

| # | Name | Type |
|----------------------------|-----------------|-------------|
| <input type="checkbox"/> 1 | Make | varchar(25) |
| <input type="checkbox"/> 2 | Model | varchar(75) |
| <input type="checkbox"/> 3 | <u>Damaged</u> | int(2) |
| <input type="checkbox"/> 4 | Serial_Number 🔑 | varchar(40) |

| # | Name | Type |
|----------------------------|-------------------|-------------|
| <input type="checkbox"/> 1 | EMPID 🔑 | varchar(10) |
| <input type="checkbox"/> 2 | ticket_assigned 🔑 | varchar(14) |
| <input type="checkbox"/> 3 | fname | varchar(50) |
| <input type="checkbox"/> 4 | lname | varchar(50) |

Normal Forms

- ▶ **employee(EMPID, TicketID, fname, lname)**
 - ▶ 2NF
 - ▶ Candidate keys: EMPID, TicketID
 - ▶ Functional dependency (EMPID → fname, lname)
 - ▶ Functional dependency (TicketID → EMPID)
- ▶ **device(Make, Model, Damaged, Serial_Number, Ticket_ID)**
 - ▶ BCNF
 - ▶ Candidate keys: Serial_Number
 - ▶ Functional dependency device(Serial_Number → Make, Model, Damaged, Ticket_ID)
- ▶ **customer(Fname, Lname, CellNum, Email, NotifyFreq, ReasonForVisit, TicketID, Preference)**
 - ▶ BCNF
 - ▶ Candidate keys: CellNum
 - ▶ Functional dependency (CellNum → Fname, Lname, Email, NotifyFreq, ReasonForVisit, TicketID, Preference)

Possible Users

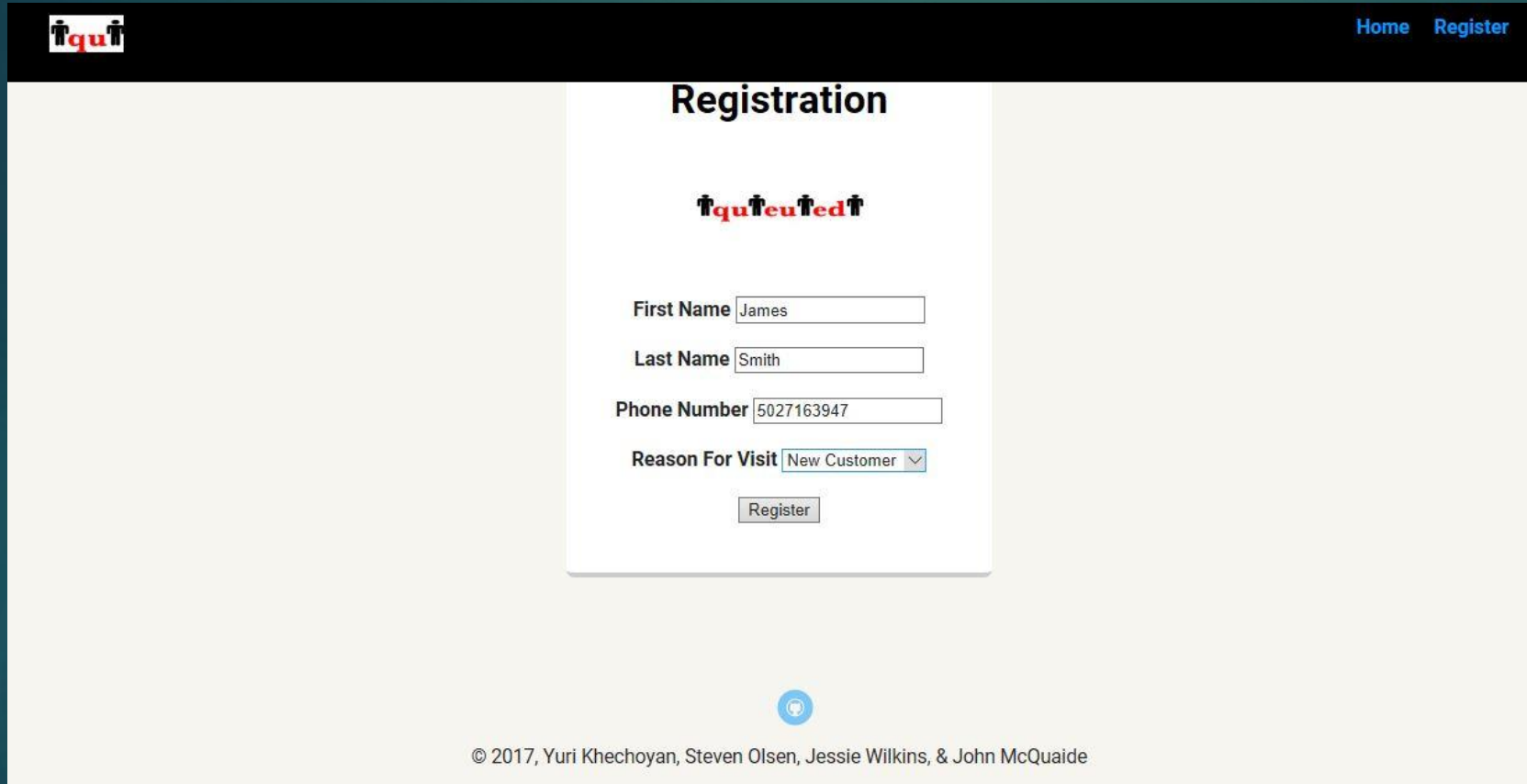
- ▶ Restaurants
- ▶ Telecommunication Retailers (AT&T, Verizon, T-Mobile, etc.)
- ▶ Cafes (Bread Co., etc.)

verizon✓

The
Cheesecake
Factory®


Panera
BREAD®

Importance of Database



The screenshot shows a web application interface with a dark blue header. On the left is a logo with the text 'tqu' and two stylized human figures. On the right are links for 'Home' and 'Register'. The main content area is white and features a central registration form. The form is titled 'Registration' and includes the 'tqu' logo. It contains four input fields: 'First Name' with the value 'James', 'Last Name' with the value 'Smith', 'Phone Number' with the value '5027163947', and 'Reason For Visit' with a dropdown menu showing 'New Customer'. A 'Register' button is located below the dropdown. At the bottom of the page, there is a small blue circular icon and a copyright notice: '© 2017, Yuri Khechoyan, Steven Olsen, Jessie Wilkins, & John McQuaide'.

Registration

tqu

First Name

Last Name

Phone Number

Reason For Visit

Register

© 2017, Yuri Khechoyan, Steven Olsen, Jessie Wilkins, & John McQuaide

- ▶ Provides easy to use UI for Management & Employees
- ▶ Gives Higher Level Permissions to Management
- ▶ Exports all information to an Excel Spreadsheet

Webpages + Queries

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 4 (5 total, Query took 0.0028 seconds.)

```
SELECT Fname, Lname, Cellnum, ReasonForVisit, TicketID FROM customer WHERE ReasonForVisit = 'Device'
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Export](#)]

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options

| Fname | Lname | Cellnum | ReasonForVisit | TicketID |
|---------|--------|-------------|----------------|---------------|
| Tricia | Welch | 6358973456 | Device | 2017/12/11-9 |
| Allie | Gai | 6360125647 | Device | 2017/12/11-3 |
| John | Smith | 6366680011 | Device | 2017/12/11-0 |
| Rachel | Stumpf | 6367781223 | Device | 2017/12/11-13 |
| Cameron | Renz | 65411004511 | Device | 2017/12/11-6 |

SELECTING SPECIFIC CUSTOMERS FROM DATABASE EXAMPLE

SELECT FNAME, LNAME, CELL NUM, REASONFORVISIT, TICKETID FROM CUSTOMER WHERE REASONFORVISIT = 'DEVICE'

⚠️ Current selection does not contain a unique column. Undo, edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 3 (4 total, Query took 0.0020 seconds.)

```
SELECT COUNT(ticket_assigned), EMPID, fname, lname FROM employee GROUP BY EMPID
```

☐ Profiling [Edit inline] [Edit] [Explains]

☐ Show all | Number of rows: 25 ▾ | Filter rows:

+ Options

| COUNT(ticket_assigned) | EMPID | fname | lname |
|------------------------|------------|--------|----------|
| 4 | A123456789 | Steven | Olson |
| 4 | B123456789 | John | McQuaide |
| 3 | C123456789 | Jessie | Wilkins |
| 3 | D123456789 | Yuri | Kechoyan |

☐ Show all | Number of rows: 25 ▾ | Filter rows:

Query results operations

**SELECTING THE
NUMBER OF TICKETS
THAT AN EMPLOYEE
HAS ASSIGNED
THEMSELVES TO (HOW
MANY CUSTOMERS
ARE THEY
INTERACTING WITH):**

```
SELECT COUNT(TICKET  
_ASSIGNED), EMPID, F  
NAME, LNAME FROM  
EMPLOYEE GROUP BY  
EMPID
```

Server: 127.0.0.1 > Database: project_db > Table: employee

Browse Structure SQL Search Insert Export Import

Show query box

✓ Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

```
SELECT fname, lname, ticket_assigned, EMPID FROM employee WHERE ticket_assigned = '2017/12/11-0'
```

Profiling [Edit inline]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

| | fname | lname | ticket_assigned | EMPID |
|---|--------|-------|-----------------|------------|
| <input type="checkbox"/> Edit Copy Delete | Steven | Olson | 2017/12/11-0 | A123456789 |

↑ ☐ Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

RETRIEVING WHICH
EMPLOYEE ASSISTED
WHICH CUSTOMER BASED
ON SPECIFIC TICKETID :

```
SELECT FNAME,  
LNAME,TICKET_ASSIGNED,  
EMPID FROM EMPLOYEE  
WHERE TICKET_ASSIGNED  
='2017/12/11/-0';
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with 'project_db' selected. The main panel shows a SQL query: `SELECT e.EMPID, count(*) as ticket_assigned FROM employee e GROUP BY e.EMPID UNION ALL SELECT 'A', AVG(T.TICKET_ASSIGNED) FROM (SELECT COUNT(*) AS TICKET_ASSIGNED FROM EMPLOYEE E2 GROUP BY E2.EMPID) T`. The results table has two columns: 'EMPID' and 'ticket_assigned'. The data rows are: A123456789 (4.0000), B123456789 (4.0000), C123456789 (3.0000), D123456789 (3.0000), and an average row 'avg =' (3.5000). The interface includes a 'Show query box' section, a warning message about unique columns, and a 'Query results operations' bar at the bottom.

| EMPID | ticket_assigned |
|------------|-----------------|
| A123456789 | 4.0000 |
| B123456789 | 4.0000 |
| C123456789 | 3.0000 |
| D123456789 | 3.0000 |
| avg = | 3.5000 |

RETRIEVING THE AVERAGE AMOUNT OF CUSTOMERS THE EMPLOYEES ASSISTS:

```
SELECT E.EMPID, COUNT(*) AS TICKET_ASSIGNED FROM EMPLOYEE E GROUP BY E.EMPID UNION ALL SELECT 'A', AVG(T.TICKET_ASSIGNED) FROM (SELECT COUNT(*) AS TICKET_ASSIGNED FROM EMPLOYEE E2 GROUP BY E2.EMPID) T
```

localhost / 127.0.0.1 / phpMyAdmin

MySQL :: MySQL 5.5 Reference Manual

localhost/phpmyadmin/tbl_sql.php?db=project_db&table=customer

Server: 127.0.0.1 » Database: project_db » Table: customer

Browse Structure SQL Search

Show query box

Showing rows 0 - 13 (14 total, Query took 0.0019 seconds)

`SELECT CellNum, COUNT(TicketID) FROM customer GROUP BY CellNum`

Show all | Number of rows: 25 | Filter rows

+ Options

| CellNum | COUNT(TicketID) |
|-------------|-----------------|
| 3141239876 | 1 |
| 3210047564 | 1 |
| 5337789981 | 1 |
| 6358973456 | 1 |
| 6360125647 | 1 |
| 6366680011 | 1 |
| 6367110012 | 1 |
| 6367781223 | 1 |
| 6367890011 | 1 |
| 6368710045 | 1 |
| 6542234400 | 1 |
| 6890011234 | 1 |
| 7890014490 | 1 |
| 65411004511 | 1 |

Show all | Number of rows: 25 | Filter rows

Console

RETRIEVING THE AMOUNT OF TIMES A CUSTOMER COMES IN AND GETS ASSISTED BY EMPLOYEE BASED ON THE AMOUNT OF TICKETIDS THE THAT ARE ASSOCIATED WITH A CUSTOMER'S CELL NUMBER:

`SELECT CELLNUM, COUNT(TICKETID) FROM CUSTOMER GROUP BY CELLNUM`

Pre-recorded Live Demo

