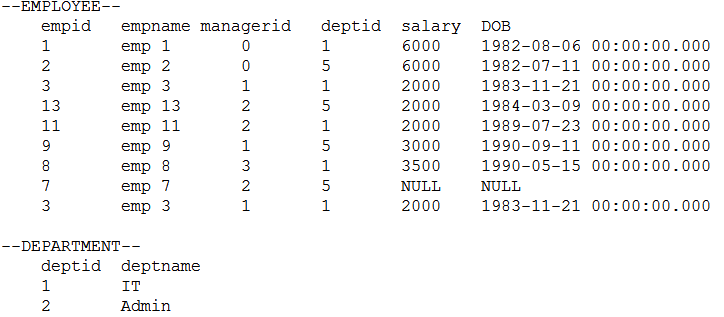
# Take-homeSQL Assignment



1. Write a SQL query to delete duplicate records from employee table which has no primary key.

Ans –

WITH DuplicateEmps AS (

SELECT EmpId, EmpName, ManagerId, DeptId, Salary, DOB, ROW\_NUMBER() OVER (

PARTITION BY EmpId, EmpName, ManagerId, DeptId, Salary

ORDER BY EmpId, EmpName, ManagerId, DeptId, Salary

) AS DuplicateCnt

FROM EMPLOYEE

);

DELETE FROM DuplicateEmps

WHERE DuplicateCnt > 1

1. Suppose there is a table called REGION with columns (regid, regname). Before you could insert any data in it you realize that you no longer need this table in your database. You execute the below query to drop this table. What will be its result? Provide explanation

IF EXISTS (SELECT TOP 1 \* FROM dbo.region)

BEGIN

DROP TABLE dbo.region;

END

Ans –

As per told in the example that there is no data present in table “Region”. Table “Region” will not get deleted but query will get successfully executed because IF EXISTS statement checks that if there is record returning from given query. If at least one record is returned from “SELECT TOP 1 \* FROM dbo.region” query then table will get dropped. But in this case table “Region” will not get drop as there is no records returned by the query.

1. Write a query to get employee details whose department id is not valid or department id is not present in the department table. There is more than one way to write this query. Which method would you least recommend to another developer?

Ans –

Way 1:

SELECT Emp.\*

FROM EMPLOYEE Emp

WHERE Emp.DeptId NOT IN (

SELECT DISTINCT Dept.DeptId

FROM DEPARTMENT Dept

)

Way 2:

SELECT Emp.\*

FROM EMPLOYEE Emp JOIN DEPARTMENT Dept

ON Emp.DeptId <> Dept.DeptId

Way 3:

SELECT Emp.\*

FROM EMPLOYEE Emp Left Join DEPARTMENT Dept

ON Emp.DeptId = Dept.DeptId

WHERE Dept.DeptId IS NULL

“Way 1” I would recommend to another developers Because NOT IN Takes longer time than JOIN used in other 2 queries.

# Take-home React Assignment

We are looking to evaluate your proficiency with React, Redux, APIs and front-end design with a small project. There will be two parts, each one hour or less to complete.

## Setup

You can use create-react-app to scaffold your application. <https://github.com/facebook/create-react-app>

## Part 1: API Integration

Given the following API endpoints:

* [1] <https://api.inquickerstaging.com/v3/winter.inquickerstaging.com/services>
* [2] <https://api.inquickerstaging.com/v3/winter.inquickerstaging.com/providers?include=locations%2Cschedules.location&page%5Bnumber%5D=1&page%5Bsize%5D=10>

### Assignment

Having fetched a list of services [1] and providers [2] from the API, create an interface that has two sections:

* A 'control' section which displays each service.
* A 'results' section which displays all provider names.

Then:

* If a service name is clicked, that service gets highlighted, and
* Only providers associated with that service gets shown in the 'results' section.

### Evaluation

We will not be evaluating front-end UI/UX for this portion. We will be primarily looking for:

* [ ] Data successfully retrieved from API
* [ ] Data stored in Redux
* [ ] Filtering functionality using Redux state
* [ ] Data correctness

## Part 2: Design and Code Quality

### Assignment

Building on the previous section, we are going to enhance the results list to include more than just the providers' names.

Create a component that shows:

* a provider's image, on the left
* their name, on the right
* subspecialties, if any, on the right, below the name

Here is an ASCII representation of the component:

|----------------------------------|

| |-------| Provider Name |

| | image | |

| | | Subspecialties |

| |-------| |

|----------------------------------|

Feel free to use a UI library such as <https://material-ui.com/> to speed up development.

### Evaluation

We will not evaluate the UI/UX of the control section (that contains the service names) so that can remain relatively unstyled. We will be looking primarily at the result section, the new component, and how the code is organized/written. The criteria:

* [ ] Application is composed of multiple components.
* [ ] Component renders well, and with correct layout, on mobile/desktop viewports.
* [ ] Data correctness, when comparing frontend results to API data.

### Submission

* Your project should be hosted in a BitBucket or GitHub repository. Please provide the direct link to the repo.
* Your readme should include any steps necessary to run the project locally.