

## Module 2 – Databases and Backend Development(Node.js)

### Node.js Exercise 6

#### Overview

In this exercise, you will create a mini robust express application for a retailer called 'PICK 'n STEAL'. Your express app will only contain their employees and their departments.

#### Instructions:

// TODO: Create and develop a fully functional API that can be used to dynamically access and query the employee data.

1. Make sure to use all the necessary dependencies are used to create your API (for routes and accessing mysql)

HINT:: Use the Database created for Exercise 4 (pick\_n\_steal) to complete this exercise. You **must** use **Thunder Client** or **Postman** to complete the CRUD operations below (always make the functions dynamic).

2. a. Create a new table called departments.

	Field	Type	Null	Key	Default	Extra
▶	department_id	int	NO	PRI	NULL	auto_increment
	department_name	varchar(50)	NO		NULL	
	location	varchar(100)	YES		NULL	

b. Alter the employees table so that it contains the new column (department\_id)

	Field	Type	Null	Key	Default	Extra
▶	employee_id	int	NO	PRI	NULL	auto_increment
	first_name	varchar(45)	NO		NULL	
	last_name	varchar(45)	NO		NULL	
	email	varchar(100)	NO		NULL	
	phone_number	varchar(15)	NO		NULL	
	department	varchar(50)	NO		NULL	
	salary	decimal(10,2)	NO		NULL	
	department_id	int	YES		NULL	

c. Now remove the department column in the employees table

	Field	Type	Null	Key	Default	Extra
▶	employee_id	int	NO	PRI	NULL	auto_increment
	first_name	varchar(45)	NO		NULL	
	last_name	varchar(45)	NO		NULL	
	email	varchar(100)	NO		NULL	
	phone_number	varchar(15)	NO		NULL	
	salary	decimal(10,2)	NO		NULL	
	department_id	int	YES		NULL	

d. Now insert the new data into the department table(department\_id, department\_name, location) and the employee table.(department\_id)

	department_id	department_name	location
▶	1	Engineering	South Africa
	2	Marketing	Europe
✱	NULL	NULL	NULL

	employee_id	first_name	last_name	email	phone_number	salary	department_id
▶	1	John	Doe	john.doe@example.com	555-1234	85000.00	1
	2	Jane	Smith	jane.smith@example.com	555-5678	92000.00	2
	3	Michael	Johnson	michael.johnson@example.com	555-8765	75000.00	2
	4	Emily	Davis	emily.davis@example.com	555-4321	78000.00	1
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

HINT: Use WORKBENCH to make the changes

3. Create a route with a path of 'employees':

- that returns all the employees in the database.
- that returns a single employee based on its primary key.
- that inserts a new employee within the database.
- that deletes an employee based on its primary key.
- that removes all employees.
- COMPULSORY BONUS QUESTION:: Create a route that returns all employees that are part of the department in South Africa.

4. Make sure that your code meets the following requirements of MVC architecture.

**Reminder:** Make sure to use Thunder Client or Postman to test your API.

### Submission:

Make sure that you submit your github link with a public repository to LMS with the completed work.. **NO ZIP FILE!!!**.

(What else should I say Gordon? 🍕🔥)