- What is wrong with this query?

```
SELECT Id, YEAR(TrialDate) AS TrialYear
FROM Payments
WHERE TrialYear <= 2015;
```

TrialYear will be assigned as the end, so you won't be able to use it as a filter on WHERE.

Instead you should use:

```
WHERE YEAR(TrialDate) <= 2015
```

- What is wrong with this query?

```
SELECT Id, TrialDate
FROM Payments
GROUP BY Id;
```

There was no aggregate function on the TrialDate column. There needs to be an aggregate function, otherwise the column should also be in the GROUP BY statement.

- What is wrong with this query?

```
SELECT UserId, AVG(Total) AS AvgOrderTotal

FROM Invoices

HAVING COUNT(OrderId) >= 1
```

Note the use of HAVING and the AVG function. There was no GROUP BY statement to accompany those statements, in other words, the SQL statement was missing.

Consider the two tables below.

Employees

Id	Name	ManagedBy
1	Jane Doe	NULL
2	Mark Smith	1
3	Sally Rogers	3

Managers

Id	Name
1	Zach Allen
2	Bill Lee
3	Sandy Kim

Write a query that retrieves all employees managed by Sandy Kim.

```
SELECT Name FROM Employees
WHERE ManagedBy = 3
```

```
SELECT Employees.Name FROM Employees
JOIN Managers ON
Employees.ManagedBy = Managers.Id
WHERE Managers.Name LIKE “Sandy Kim”
```

Consider the two tables below.

Employees

Id	Name	ManagedBy
1	Jane Doe	NULL
2	Mark Smith	1
3	Sally Rogers	3

Managers

Id	Name
1	Zach Allen
2	Bill Lee
3	Sandy Kim

Write a query that retrieves all employees that have no manager.

- Write a query that retrieves all employees that have no manager.

```
SELECT Name FROM Employees
WHERE ManagedBy Is Null
```