Part 1: SQL Joins

1. INNER JOIN

An INNER JOIN will return only the rows where there is a match in both tables.

sql

Sample code

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM Employee e
INNER JOIN Department d ON e.department_id = d.department_id;
```

Result:

Sample code

```
employee_id  first_name  last_name  department_name
1            John        Doe        HR
2            Jane        Smith      Sales
3            Mike        Johnson    IT
4            Emily       Davis      HR
```

2. LEFT OUTER JOIN

A LEFT OUTER JOIN will return all the rows from the left table (Employee), and the matched rows from the right table (Department). If there is no match, NULLs are returned for columns from the right table.

sql

Sample code

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
```

```
FROM Employee e
LEFT OUTER JOIN Department d ON e.department_id = d.department_id;
```

Result:

Sample code

```
employee_id  first_name  last_name  department_name
1            John        Doe        HR
2            Jane        Smith      Sales
3            Mike        Johnson    IT
4            Emily       Davis      HR
```

3. RIGHT OUTER JOIN

A RIGHT OUTER JOIN will return all the rows from the right table (Department), and the matched rows from the left table (Employee). If there is no match, NULLs are returned for columns from the left table.

sql

Sample code

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM Employee e
RIGHT OUTER JOIN Department d ON e.department_id = d.department_id;
```

Result:

arduino

Sample code

```
employee_id  first_name  last_name  department_name
```

1 John Doe HR

4 Emily Davis HR

2 Jane Smith Sales

3 Mike Johnson IT

NULL NULL NULL Marketing

4. FULL OUTER JOIN

A FULL OUTER JOIN returns all rows when there is a match in either left (Employee) or right (Department) table. If there is no match, NULLs are returned for columns from the table that lacks a match.

sql

Sample code

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM Employee e
FULL OUTER JOIN Department d ON e.department_id = d.department_id;
```

Result:

arduino

Sample code

employee_id first_name last_name department_name

1 John Doe HR

4 Emily Davis HR

2 Jane Smith Sales

3 Mike Johnson IT

NULL NULL NULL Marketing

Part 2: Finding Duplicate Records

Let's consider the Employee table with email:

employee_id first_name last_name email

1 John Doe john.doe@example.com

2 Jane Smith jane.smith@example.com

3 John Doe john.doe@example.com

4 Emily Davis emily.davis@example.com

1. Based on first_name

sql

Sample code

```
SELECT first_name, COUNT(*)
FROM Employee
GROUP BY first_name
HAVING COUNT(*) > 1;
```

Result:

scss

Sample code

first_name COUNT(*)

John 2

2. Based on email

sql

Sample code

```
SELECT email, COUNT(*)
FROM Employee
GROUP BY email
HAVING COUNT(*) > 1;
```

Result:

scss

Sample code

```
email               COUNT(*)
john.doe@example.com  2
```

3. Based on first_name and last_name

sql

Sample code

```
SELECT first_name, last_name, COUNT(*)
FROM Employee
GROUP BY first_name, last_name
HAVING COUNT(*) > 1;
```

Result:

scss

Sample code

first_name last_name COUNT(*)

John Doe 2

4. Based on first_name and email

sql

Sample code

```
SELECT first_name, email, COUNT(*)
FROM Employee
GROUP BY first_name, email
HAVING COUNT(*) > 1;
```

Result:

scss

Sample code

first_name email COUNT(*)

John john.doe@example.com 2

These queries help identify the duplicate records based on different criteria in the Employee table.