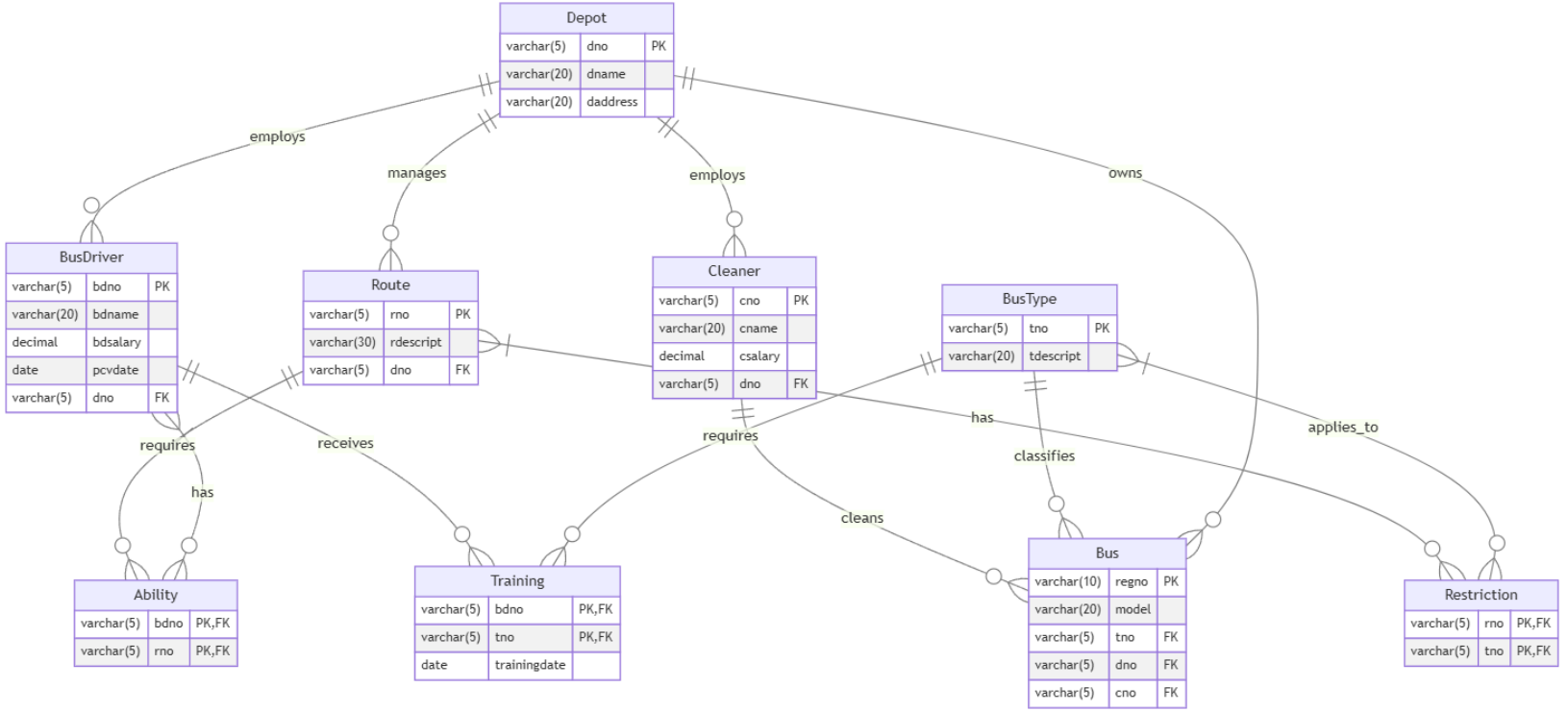
“Bus_Depots” Database

The original code provided was designed for an Oracle database. As the tutorial follow was for MySQL the script was converted accordingly before beginning the exercise. Adjustments included replacing VARCHAR2 with VARCHAR, using NULL for nullable fields instead of empty strings, and standardising date formats to YYYY-MM-DD.

Tools Used:

https://onecompiler.com/ https://pinetools.com/syntax-highlighter https://mermaid.live/



SHOW TABLES;

ability
bus
busdriver
bustype
cleaner
depot
restriction
route
training

SELECT * FROM Restriction;

rno	tno
10	1
11	1
6	1
7	1
10	2
11	2
6	2
7	2
10	3
11	3
6	3
8	3
10	4
11	4
6	4
8	4

SELECT * FROM Training;

bdno	tno	trainingdate
001	1	2006-01-09
001	2	2006-01-09
006	2	2006-02-09
007	1	2006-02-09
007	2	2006-02-09
007	3	2006-03-09
008	2	2006-03-09
008	3	2006-03-09
008	4	2006-04-09
009	3	2006-04-09
009	4	2006-05-09
011	1	2006-05-09
011	2	2006-05-09
011	3	2006-06-09
011	4	2006-06-09
011	5	2006-06-09

SELECT * FROM Ability;

bdno	rno
007	10
008	10
008	11
001	6
007	6
001	7
009	7
001	8

SELECT * FROM Cleaner;

cno	cname	csalary	dno
110	John	2550.00	101
111	Jean	2500.00	101
112	Betty	2400.00	102
113	Vince	2800.00	102
114	Jay	3000.00	102
115	Doug	2000.00	102
116	Geeta	4000.00	NULL

SELECT * FROM BusType;

tno	tdescript
1	double-decker
2	metrobus
3	midibus
4	bendy bus
5	open top

SELECT * FROM Route;

rno	rdescript	dno
10	Tottenham/Angel	102
11	Islington/Highgate	102
6	Camden/Golders Green	101
7	Finchley/Tottenham	101
8	Hendon/Muswell Hill	101

SELECT * FROM BusDriver;

bdno	bdname	bdsalary	pcvdate	dno
001	Jane Brown	1800.00	1985-02-09	101
006	Sally Smith	1750.00	1996-03-09	NULL
007	James Bond	1500.00	1999-01-09	102
008	Maggie May	2200.00	2000-01-09	102
009	Jack Jones	1400.00	2001-08-09	101
010	Peter Piper	3500.00	2004-06-09	104
011	John Peel	2000.00	2005-02-09	102

SELECT * FROM Depot;

dno	dname	daddress
101	Holloway	Camden Road
102	Hornsey	High Road
104	Islington	Upper Street

SELECT * FROM Bus;

regno	model	tno	dno	cno
A123ABC	Routemaster	1	101	110
D345GGG	Volvo 8500	1	101	112
D678FGH	Volvo 8700	2	101	110
H259IJK	Daf SB220	3	102	114
P200IJK	Mercedes 709D	2	102	113
P300RTY	Mercedes Citaro	4	102	113
R678FDS	Daf SB220	1	NULL	110

Advanced SQL 1

1. (Project, restrict) List all drivers (number and name) who have a salary of less than 1,800.

```
SELECT bdno, bdname
FROM BusDriver
WHERE bdsalary < 1800;
```

```
+------+-------------+
| bdno | bdname      |
+------+-------------+
| 006  | Sally Smith |
| 007  | James Bond  |
| 009  | Jack Jones  |
+------+-------------+
```

2. (Conditional operator LIKE) List all bus drivers (number and name) whose name begins with J.

```
SELECT bdno, bdname
FROM BusDriver
WHERE bdname LIKE 'J%';
```

```
+------+------------+
| bdno | bdname     |
+------+------------+
| 001  | Jane Brown |
| 007  | James Bond |
| 009  | Jack Jones |
| 011  | John Peel  |
+------+------------+
```

3. (Conditional operator BETWEEN) List all bus drivers' details for those drivers who have a salary between 2,000 and 4,000.

```
SELECT *
FROM BusDriver
WHERE bdsalary BETWEEN 2000 AND 4000;
```

```
+------+-------------+----------+------------+------+
| bdno | bdname      | bdsalary | pcvdate    | dno  |
+------+-------------+----------+------------+------+
| 008  | Maggie May  |  2200.00 | 2000-01-09 | 102  |
| 010  | Peter Piper |  3500.00 | 2004-06-09 | 104  |
| 011  | John Peel   |  2000.00 | 2005-02-09 | 102  |
+------+-------------+----------+------------+------+
```

4. (And) List all buses (registration number and model) of type 2 which are not based at depot 101.

```
SELECT regno, model
FROM Bus
WHERE tno = '2' AND dno != '101';
```

```
+---------+---------------+
| regno   | model         |
+---------+---------------+
| P200IJK | Mercedes 709D |
+---------+---------------+
```

5. (Or) List buses (all details) which are either Volvo models or Mercedes models. What is the output when you change OR to AND?

```
SELECT *
FROM Bus
WHERE model LIKE 'Volvo%' OR model LIKE 'Mercedes%';
```

```
+---------+-----------------+------+------+------+
| regno   | model           | tno  | dno  | cno  |
+---------+-----------------+------+------+------+
| D345GGG | Volvo 8500      | 1    | 101  | 112  |
| D678FGH | Volvo 8700      | 2    | 101  | 110  |
| P200IJK | Mercedes 709D   | 2    | 102  | 113  |
| P300RTY | Mercedes Citaro | 4    | 102  | 113  |
+---------+-----------------+------+------+------+
```

When you change OR to AND, the query will only return buses that are both Volvo and Mercedes models, which is should not possible.

6. (Controlling duplicates using DISTINCT) List all depot numbers in the bus table. Now eliminate all duplicates.

```
SELECT DISTINCT dno
FROM Bus;
```

```
+------+
| dno  |
+------+
| NULL |
| 101  |
| 102  |
+------+
```

7. (Two table Join – Inner Join) List all cleaners (number and name) with the name an address of their depot, but only for those cleaners located at a depot.

```
SELECT c.cno, c.cname, d.dname, d.daddress
FROM Cleaner c
INNER JOIN Depot d ON c.dno = d.dno;
```

```
+-----+-------+----------+-------------+
| cno | cname | dname    | daddress    |
+-----+-------+----------+-------------+
| 110 | John  | Holloway | Camden Road |
| 111 | Jean  | Holloway | Camden Road |
| 112 | Betty | Hornsey  | High Road   |
| 113 | Vince | Hornsey  | High Road   |
| 114 | Jay   | Hornsey  | High Road   |
| 115 | Doug  | Hornsey  | High Road   |
+-----+-------+----------+-------------+
```

8. (Three table Join) List bus drivers (number and name) and the bus types (description) for which each bus driver has had training.

```
SELECT bd.bdno, bd.bdname, bt.tdescript
FROM BusDriver bd
INNER JOIN Training t ON bd.bdno = t.bdno
INNER JOIN BusType bt ON t.tno = bt.tno;
```

```
+------+-------------+---------------+
| bdno | bdname      | tdescript     |
+------+-------------+---------------+
| 001  | Jane Brown  | double-decker |
| 007  | James Bond  | double-decker |
| 011  | John Peel   | double-decker |
| 001  | Jane Brown  | metrobus      |
| 006  | Sally Smith | metrobus      |
| 007  | James Bond  | metrobus      |
| 008  | Maggie May  | metrobus      |
| 011  | John Peel   | metrobus      |
| 007  | James Bond  | midibus       |
| 008  | Maggie May  | midibus       |
| 009  | Jack Jones  | midibus       |
| 011  | John Peel   | midibus       |
| 008  | Maggie May  | bendy bus     |
| 009  | Jack Jones  | bendy bus     |
| 011  | John Peel   | bendy bus     |
| 011  | John Peel   | open top      |
+------+-------------+---------------+
```

9. (Four table Join) List all cleaners (number and name), the name of their depot and the bus registration numbers with the type of bus that they are responsible for.

```
SELECT c.cno, c.cname, d.dname, b.regno, bt.tdescript
FROM Cleaner c
INNER JOIN Depot d ON c.dno = d.dno
INNER JOIN Bus b ON c.cno = b.cno
INNER JOIN BusType bt ON b.tno = bt.tno;
```

```
+-----+-------+----------+---------+---------------+
| cno | cname | dname    | regno   | tdescript     |
+-----+-------+----------+---------+---------------+
| 110 | John  | Holloway | A123ABC | double-decker |
| 110 | John  | Holloway | D678FGH | metrobus      |
| 110 | John  | Holloway | R678FDS | double-decker |
| 112 | Betty | Hornsey  | D345GGG | double-decker |
| 113 | Vince | Hornsey  | P200IJK | metrobus      |
| 113 | Vince | Hornsey  | P300RTY | bendy bus     |
| 114 | Jay   | Hornsey  | H259IJK | midibus       |
+-----+-------+----------+---------+---------------+
```

10. (Outer join) Rewrite question 7 as an outer join. Describe the query in English. Now list all cleaners (number and name), the name of their depot and the bus registration numbers with the type of bus that they are responsible for, including those cleaners who are not assigned to a bus or a depot.

```
SELECT c.cno, c.cname, d.dname, d.daddress,
b.regno, bt.tdescript
FROM Cleaner c
LEFT OUTER JOIN Depot d ON c.dno = d.dno
LEFT OUTER JOIN Bus b ON c.cno = b.cno
LEFT OUTER JOIN BusType bt ON b.tno = bt.tno;
```

```
+-----+-------+----------+-------------+---------+---------------+
| cno | cname | dname    | daddress    | regno   | tdescript     |
+-----+-------+----------+-------------+---------+---------------+
| 110 | John  | Holloway | Camden Road | A123ABC | double-decker |
| 110 | John  | Holloway | Camden Road | D678FGH | metrobus      |
| 110 | John  | Holloway | Camden Road | R678FDS | double-decker |
| 111 | Jean  | Holloway | Camden Road | NULL    | NULL          |
| 112 | Betty | Hornsey  | High Road   | D345GGG | double-decker |
| 113 | Vince | Hornsey  | High Road   | P200IJK | metrobus      |
| 113 | Vince | Hornsey  | High Road   | P300RTY | bendy bus     |
| 114 | Jay   | Hornsey  | High Road   | H259IJK | midibus       |
| 115 | Doug  | Hornsey  | High Road   | NULL    | NULL          |
| 116 | Geeta | NULL     | NULL        | NULL    | NULL          |
+-----+-------+----------+-------------+---------+---------------+
```

This query lists all cleaners along with the name and address of their depot and the bus registration numbers with the type of bus they are responsible for. If a cleaner is not assigned to a depot or a bus, the corresponding fields will be NULL. The LEFT OUTER JOIN ensures that cleaners who have no assigned depot or bus are still included in the result set.

Advanced SQL 2

1. (Built-in functions) Find the maximum, minimum and average driver's salary.

```
SELECT
    MAX(bdsalary) AS MaxSalary,
    MIN(bdsalary) AS MinSalary,
    AVG(bdsalary) AS AvgSalary
FROM BusDriver;
```

```
+-----------+-----------+-------------+
| MaxSalary | MinSalary | AvgSalary   |
+-----------+-----------+-------------+
|   3500.00 |   1400.00 | 2021.428571 |
+-----------+-----------+-------------+
```

2. (Built-in functions) Count the number of drivers who are working for Middlesex Transport currently. Change the column heading in the result to make it 'friendly'.

```
SELECT COUNT(bdno) AS 'Number of Drivers at Middlesex Transport'
FROM BusDriver
WHERE dno IS NOT NULL;
```

```
+-----------------------------------------+
| Number of Drivers at Middlesex Transport |
+-----------------------------------------+
|                                       6 |
+-----------------------------------------+
```

3. (Use a subquery to answer this question.) Find route information (route number and description) for all routes which connect to the Holloway Depot.

```
SELECT r.rno, r.rdescript
FROM Route r
WHERE r.dno IN (SELECT dno FROM Depot WHERE dname = 'Holloway');
```

```
+-----+----------------------+
| rno | rdescript            |
+-----+----------------------+
| 6   | Camden/Golders Green |
| 7   | Finchley/Tottenham   |
| 8   | Hendon/Muswell Hill  |
+-----+----------------------+
```

4. Now try question 3 with a Join.

```
SELECT r.rno, r.rdescript
FROM Route r
INNER JOIN Depot d ON r.dno = d.dno
WHERE d.dname = 'Holloway';
```

```
+-----+----------------------+
| rno | rdescript            |
+-----+----------------------+
| 6   | Camden/Golders Green |
| 7   | Finchley/Tottenham   |
| 8   | Hendon/Muswell Hill  |
+-----+----------------------+
```

5. (Null) List bus details for any bus which has not been assigned to a depot.

```
SELECT *
FROM Bus
WHERE dno IS NULL;
```

```
+---------+-----------+------+------+------+
| regno   | model     | tno  | dno  | cno  |
+---------+-----------+------+------+------+
| R678FDS | Daf SB220 | 1    | NULL | 110  |
+---------+-----------+------+------+------+
```

6. ('Not in') List all drivers (name and number) who are on the system but are not yet responsible for a route.

```
SELECT bdno, bdname
FROM BusDriver
WHERE bdno NOT IN (SELECT bdno FROM Ability);
```

```
+------+-------------+
| bdno | bdname      |
+------+-------------+
| 006  | Sally Smith |
| 010  | Peter Piper |
| 011  | John Peel   |
+------+-------------+
```

7. (‘group by’) List each depot name and the average salary for drivers working at the depot.

```
SELECT d.dname, AVG(bd.bdsalary) AS AvgSalary
FROM BusDriver bd
INNER JOIN Depot d ON bd.dno = d.dno
GROUP BY d.dname;
```

```
+-----------+-------------+
| dname     | AvgSalary   |
+-----------+-------------+
| Holloway  | 1600.000000 |
| Hornsey   | 1900.000000 |
| Islington | 3500.000000 |
+-----------+-------------+
```

8. (‘group by having’) List each depot by name and count the number of bus drivers who are assigned to each, for depots with more than one driver.

```
SELECT d.dname, COUNT(bd.bdno) AS DriverCount
FROM Depot d
INNER JOIN BusDriver bd ON d.dno = bd.dno
GROUP BY d.dname
HAVING COUNT(bd.bdno) > 1;
```

```
+----------+-------------+
| dname    | DriverCount |
+----------+-------------+
| Holloway |           2 |
| Hornsey  |           3 |
+----------+-------------+
```

9. (‘ group by’ plus join) For each cleaner responsible for buses of bus type double decker or minibus, list his/her name and number, and find the total number for which each cleaner is responsible.

```
SELECT c.cno, c.cname, COUNT(b.regno) AS TotalBuses
FROM Cleaner c
INNER JOIN Bus b ON c.cno = b.cno
INNER JOIN BusType bt ON b.tno = bt.tno
WHERE bt.tdescript IN ('double-decker', 'minibus')
GROUP BY c.cno, c.cname;
```

```
+-----+-------+------------+
| cno | cname | TotalBuses |
+-----+-------+------------+
| 110 | John  |          2 |
| 112 | Betty |          1 |
+-----+-------+------------+
```

10. (‘order by’)
    a. List all drivers (name and number) and their routes (number and description, order by driver number.

```
SELECT bd.bdno, bd.bdname, r.rno, r.rdescript
FROM BusDriver bd
INNER JOIN Ability a ON bd.bdno = a.bdno
INNER JOIN Route r ON a.rno = r.rno
ORDER BY bd.bdno;
```

```
+------+------------+-----+----------------------+
| bdno | bdname     | rno | rdescript            |
+------+------------+-----+----------------------+
| 001  | Jane Brown | 6   | Camden/Golders Green |
| 001  | Jane Brown | 7   | Finchley/Tottenham   |
| 001  | Jane Brown | 8   | Hendon/Muswell Hill  |
| 007  | James Bond | 10  | Tottenham/Angel      |
| 007  | James Bond | 6   | Camden/Golders Green |
| 008  | Maggie May | 10  | Tottenham/Angel      |
| 008  | Maggie May | 11  | Islington/Highgate   |
| 009  | Jack Jones | 7   | Finchley/Tottenham   |
+------+------------+-----+----------------------+
```

    b. Now order by route description within driver number.

```
SELECT bd.bdno, bd.bdname, r.rno, r.rdescript
FROM BusDriver bd
INNER JOIN Ability a ON bd.bdno = a.bdno
INNER JOIN Route r ON a.rno = r.rno
ORDER BY bd.bdno, r.rdescript;
```

```
+------+------------+-----+----------------------+
| bdno | bdname     | rno | rdescript            |
+------+------------+-----+----------------------+
| 001  | Jane Brown | 6   | Camden/Golders Green |
| 001  | Jane Brown | 7   | Finchley/Tottenham   |
| 001  | Jane Brown | 8   | Hendon/Muswell Hill  |
| 007  | James Bond | 6   | Camden/Golders Green |
| 007  | James Bond | 10  | Tottenham/Angel      |
| 008  | Maggie May | 11  | Islington/Highgate   |
| 008  | Maggie May | 10  | Tottenham/Angel      |
| 009  | Jack Jones | 7   | Finchley/Tottenham   |
+------+------------+-----+----------------------+
```