



DATABASES AND SQL PROGRAMMING LANGUAGE

BY CLAIRE GIACUZZO



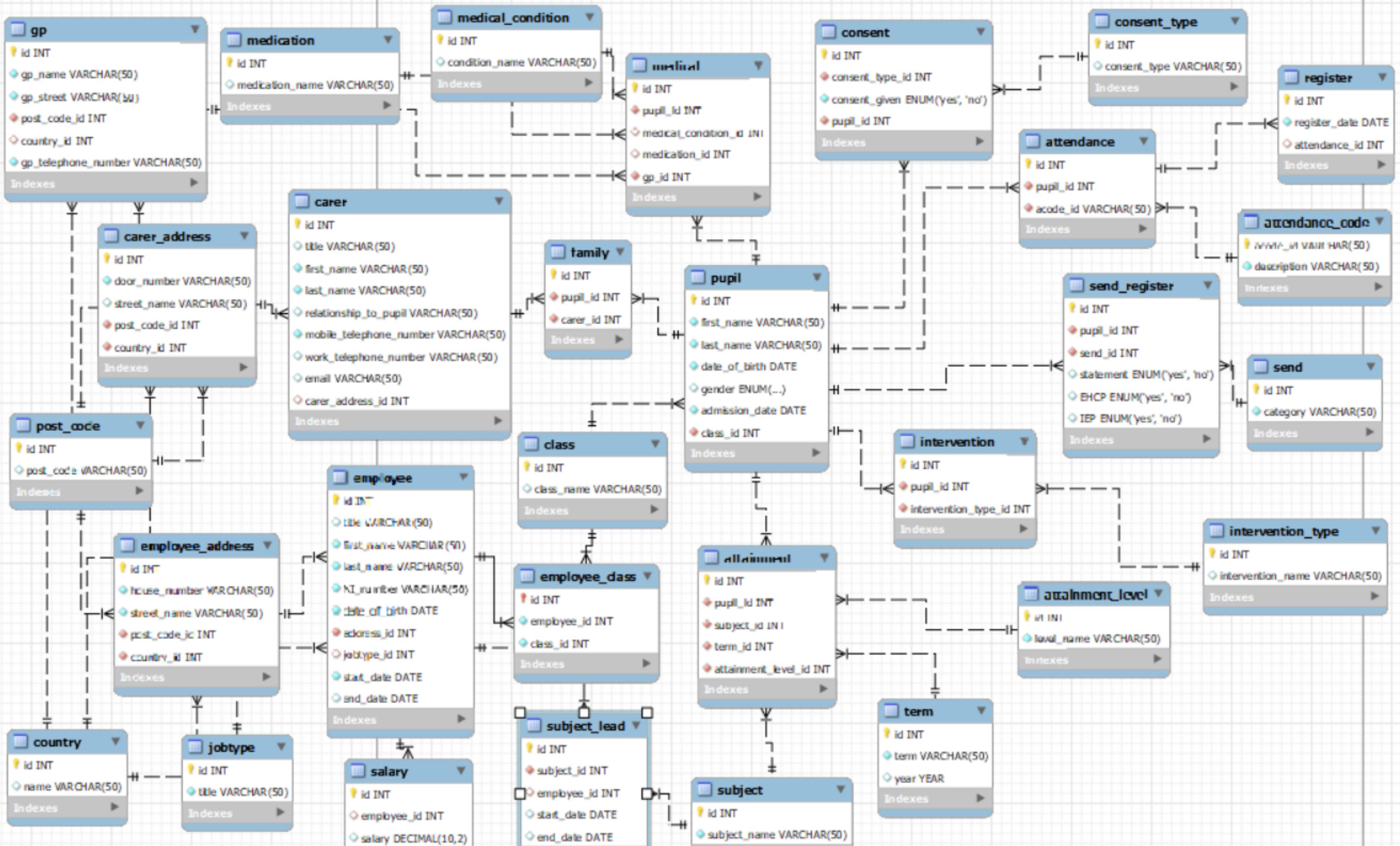
CodeFirstGirls

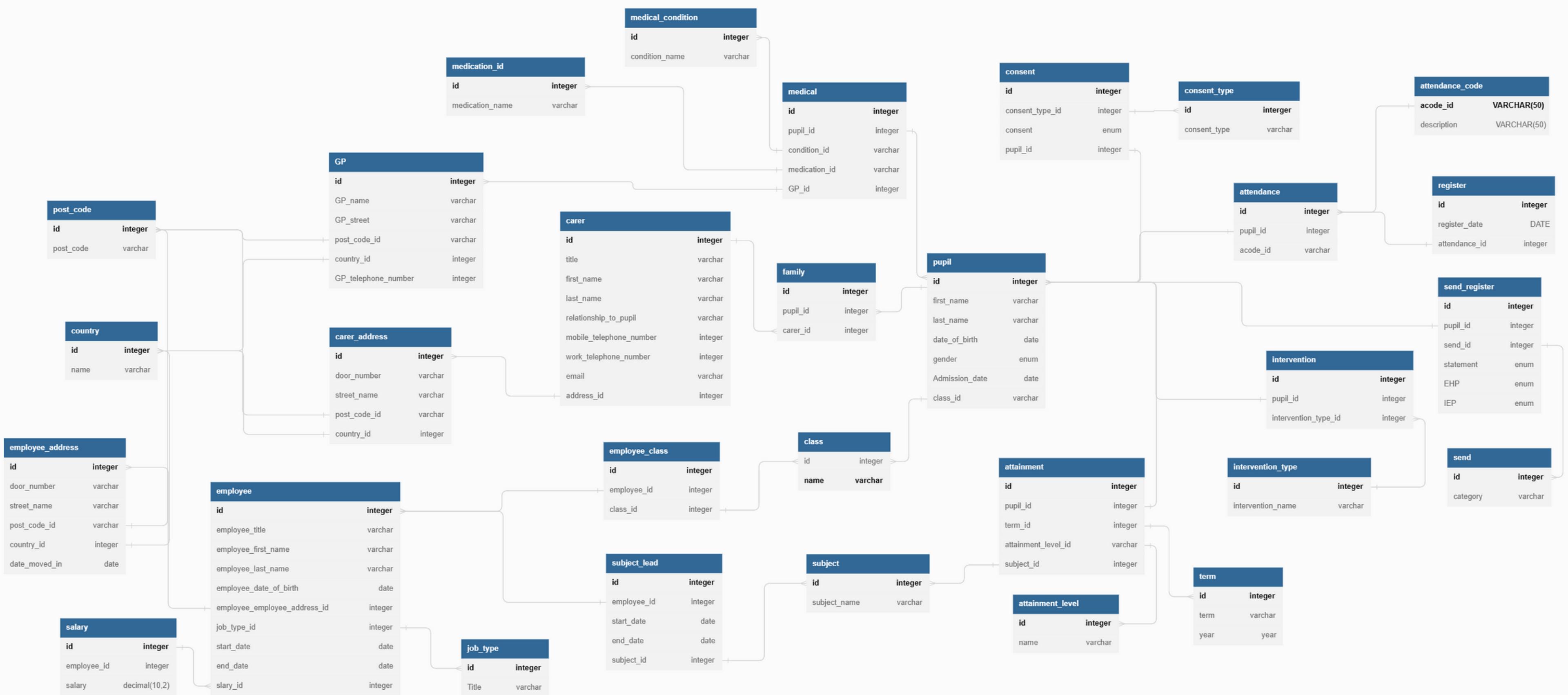


SCHOOL

CREATING A DATABASE FOR A







NORMALISATION

First Normal Form (1NF)

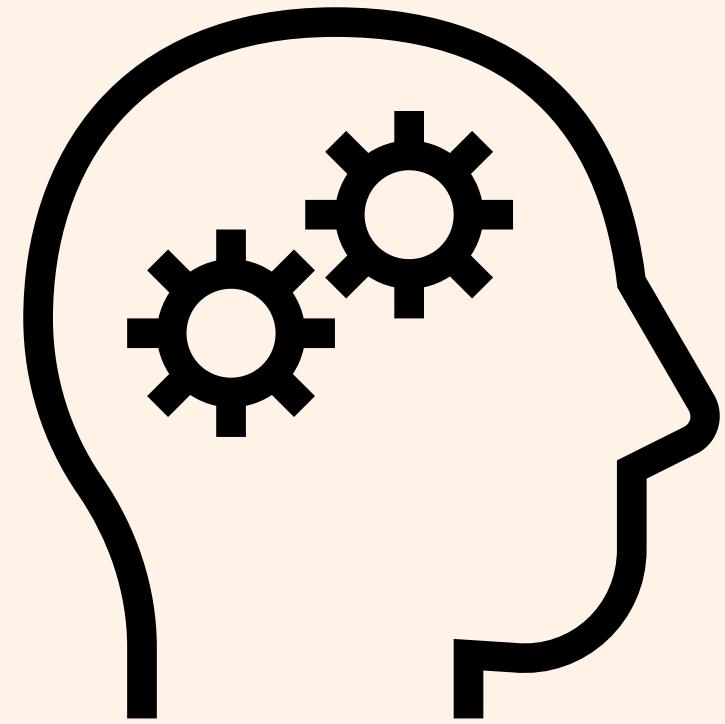
- single cell has only one value (atomicity)
- primary key for identification
- no duplicate rows or columns (carer1, carer2 etc)
- each column has only one value for each row

Second Normal Form (2NF)

- it's in 1NF
- no partial dependency - all non-key attributes are fully dependent on the primary key

Third Normal Form (3NF)

- no non-prime attribute is dependent on another non-prime attribute



Learnt

DON'T RUSH THE
NORMALISATION
PROCESS!!



Challenges

Normalisation....
too far v not enough

SQL QUERIES



MY SQL

-- Prepare an example query with group by and having to demonstrate how to extract data from your DB for analysis --

SELECT

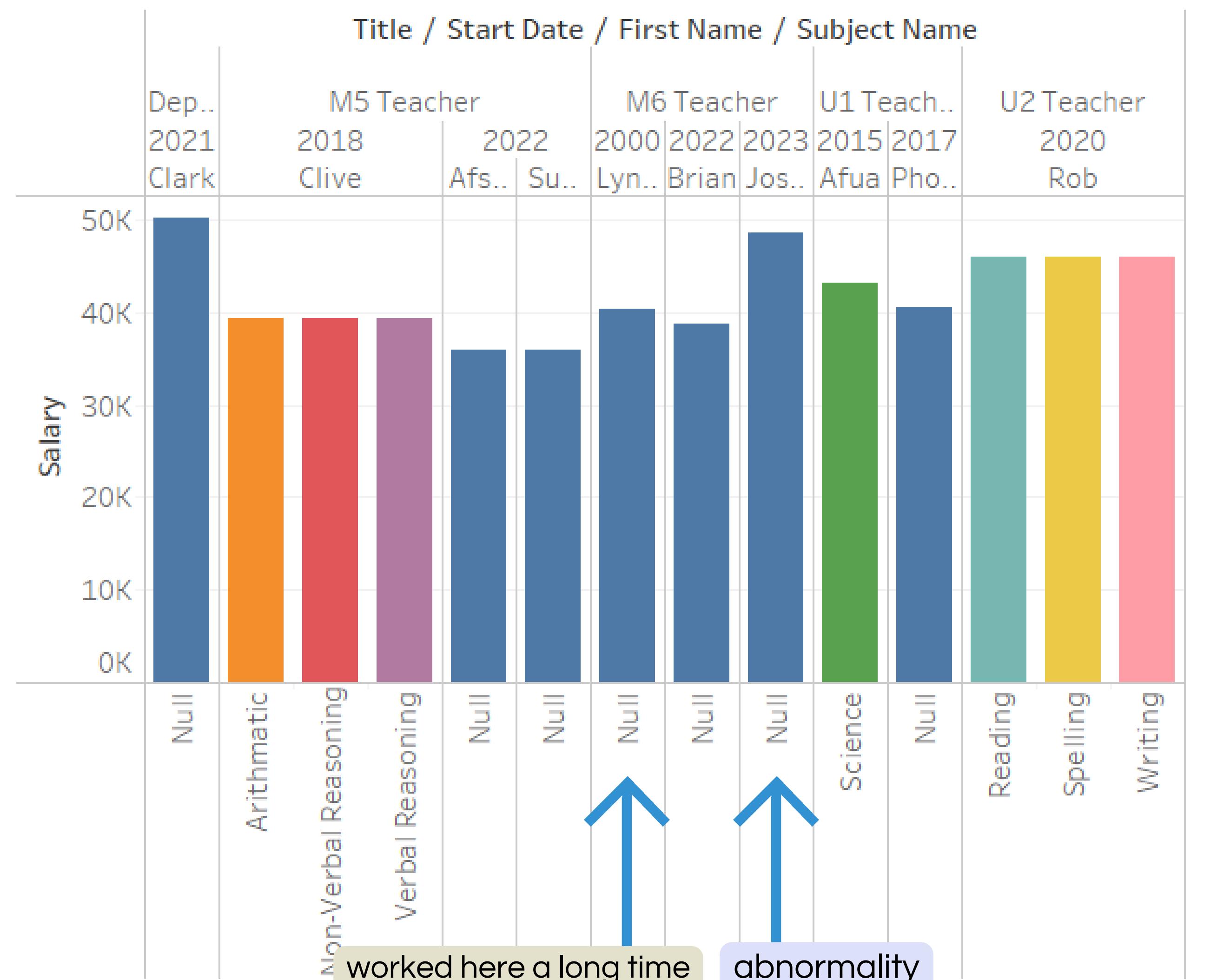
```
e.id,  
e.first_name,  
e.last_name,  
e.start_date,  
e.salary,  
j.title,  
sl.subject_id,  
su.subject_name
```

```
FROM employee AS e  
LEFT JOIN subject_lead AS sl  
    ON sl.employee_id = e.id  
LEFT JOIN subject AS su  
    ON s.id = sl.subject_id  
LEFT JOIN salary AS sl  
ON s.employee_id = e.id  
INNER JOIN jobtype AS j  
    ON j.id = e.jobtype_id  
WHERE j.title LIKE '%teacher%'  
GROUP BY 5, 1, 2, 3, 4, 6, 7  
HAVING e.salary > (SELECT AVG(salary) FROM employee)  
ORDER BY salary DESC;
```

Are **teachers** on the same banding earning the same wage? If not why? Who's earning above average and why?



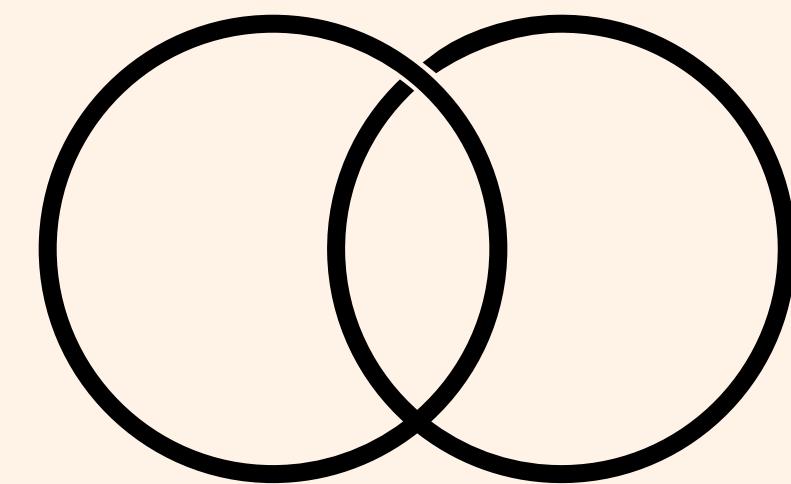
id	first_name	last_name	start_date	salary	title	subject_name
8	Clark	Jackson	2021-09-01	50122.00	Deputy Head Teacher	NULL
15	Joshua	Smith	2023-01-01	48625.00	M6 Teacher	NULL
7	Rob	Mussels	2020-09-01	46000.00	U2 Teacher	Reading
7	Rob	Mussels	2020-09-01	46000.00	U2 Teacher	Spelling
7	Rob	Mussels	2020-09-01	46000.00	U2 Teacher	Writing
5	Afua	Nairne	2015-09-01	43188.00	U1 Teacher	Science
14	Phoebe	Lile	2017-09-01	40625.00	U1 Teacher	NULL
2	Lynsey	Black	2000-09-01	40320.00	M6 Teacher	NULL
13	Clive	Hudson	2018-09-01	39290.00	M5 Teacher	Arithmatic
	Clive	Hudson	2018-09-01	39290.00	M5 Teacher	Non-Verbal Reasoning
13	Clive	Hudson	2018-09-01	39290.00	M5 Teacher	Verbal Reasoning
12	Brian	Thomas	2022-09-01	38810.00	M6 Teacher	NULL
1	Afshan	Ali	2022-09-01	35990.00	M5 Teacher	NULL
11	Sumra	Hussain	2022-01-01	35990.00	M5 Teacher	NULL



-- Using any type of the joins create a view that combines multiple tables in a logical way --

```
CREATE VIEW vw_pupil_absence
AS
SELECT
    p.first_name AS Pupil_Name,
    p.last_name AS Pupil_Surname,
    c.class_name AS class,
    cr.first_name AS Carer_Name,
    cr.last_name AS Carer_Surname,
    cr.relationship_to_pupil,
    cr.mobile_telephone_number AS mobile,
    cr.work_telephone_number AS work
FROM pupil AS p
LEFT JOIN class AS c
    ON c.id = p.class_id
LEFT JOIN carer1 AS c1
    ON c1.id = p.id
INNER JOIN attendance AS a
    ON a.pupil_id = p.id
LEFT JOIN register AS r
    ON r.attendance_id = a.id
WHERE a.acode_id = 'N'
    AND r.register_date = '2022-01-02'
GROUP BY 3, 1, 2, 4, 5, 6, 7, 8, 9;
```

Identify all pupils who are absent today and call their carer to find out the reason.



```
SELECT *
FROM vw_pupil_absence;
```

Pupil_Name	Pupil_Surname	class	Carer_Name	Carer_Surname	relationship_to_pupil	mobile	work
Robin	Dale	3AN	Claire	Dale	Foster Carer	07956371094	2045678394
Abrar	Ali	3AN	Ali	Dale	Father	07956371094	02045678394
Seb	Munroe	3AN	Clarck	Munroe	Father	04598371094	02011228394
Ruby	Greenup	1LB	Jess	Greenup	Mother	07367771094	02039908394

```

CREATE VIEW vw_sickness_injury
AS
SELECT
    p.first_name AS Pupil_Name,
    p.last_name AS Pupil_Surname,
    c.class_name AS class,
    cr.first_name AS Carer_Name,
    cr.last_name AS Carer_Surname,
    cr.relationship_to_pupil,
    cr.mobile_telephone_number AS mobile,
    cr.work_telephone_number AS work,
    ct.consent_type,
    co.consent_given AS consent,
    mc.condition_name AS medical,
    me.medication_name
FROM pupil AS p
LEFT JOIN class AS c
    ON c.id = p.class_id
INNER JOIN carer AS cr
    ON cr.id = p.id
LEFT JOIN consent AS co
    ON co.pupil_id = p.id
LEFT JOIN consent_type AS ct
    ON ct.id = co.consent_type_id
LEFT JOIN medical AS m
    ON m.id = p.id
LEFT JOIN medical_condition AS mc
    ON mc.id = m.medical_condition_id
LEFT JOIN medication AS me
    ON me.id = m.medication_id
WHERE p.first_name = 'Jessie'
    AND p.last_name = 'Jackson'
    AND ct.consent_type = 'calpol';

```

-- Example number 2 -- --joins--

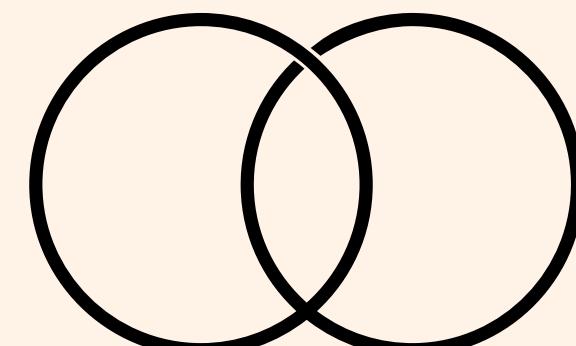
A child comes to the office with a very high temperature... We need to see their medical records to check whether they have any medical conditions. This child is prone to 'febrile convulsions' so if their temperature is high we must remove clothing, keep cool, give calpol and call their carer. We must check we have consent to give the calpol and find their carers number.

```

SELECT *
FROM vw_sickness_injury;

```

Pupil_Name	Pupil_Surname	class	Carer_Name	Carer_Surname	relationship_to_pupil	mobile	work	consent_type	consent	medical
Jessie	Jackson	3AN	Beth	Jackson	Mother	08766371094	02090878394	calpol	yes	febrile convul



-- In your database, create a stored function that can be applied to a query in your DB --

Number of years an employee has worked at the school...

```
DELIMITER //
```

```
CREATE FUNCTION no_of_years(date1 DATE)
```

```
RETURNS INT
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE date2 DATE;
```

```
    SELECT CURRENT_DATE() INTO date2;
```

```
    RETURN YEAR(date2)-YEAR(date1);
```

```
END //
```

```
DELIMITER ;
```

```
SELECT
```

```
    id, first_name, last_name, no_of_years(start_date) AS years
```

```
FROM employee;
```

id	first_name	last_name	years
1	Afshan	Ali	1
2	Lynsey	Black	23
3	Sam	Dawes	4
4	Aoife	Halsey	1
5	Afua	Nairne	8
6	Natalie	Jones	11
7	Rob	Mussels	3
8	Clark	Jackson	2
9	Isabella	Mabeka	3
10	Gina	Clarkson	9
11	Sumra	Hussain	1
12	Brian	Thomas	3
13	Clive	Hudson	5
14	Phoebe	Lile	6
15	Joshua	Smith	0

-- In your database, create a stored function that can be applied to a query in your DB --

Number of years a pupil has been at the school...

```
SELECT  
    id, first_name, last_name, no_of_years(start_date) AS years  
FROM pupil;
```

	id	first_name	last_name	years
▶	1	Robin	Dale	5
	2	Evie	Jones	5
	3	Abrar	Ali	5
	4	Jessie	Jackson	5
	5	Seb	Munroe	5
	6	Amelia	Sabbit	5
	7	Ruby	Greenup	2
	8	Kay	Arceus	2
	9	Ruby	Arceus	2
	10	Phoebe	Peters	2
	14	Zak	Greenup	1

-- Prepare an example query with a subquery to demonstrate how to extract data from your DB for analysis --

Fire in 'Hale End Road' - find out which pupils may have been effected.

```
SELECT
    p.first_name,
    p.last_name,
    c.class_name
FROM
    pupil AS p
LEFT JOIN class AS c
    ON c.id = p.class_id
WHERE
    p.id IN (
        SELECT
            c.id
        FROM carer AS c
        WHERE c.id IN (
            SELECT ca.id
            FROM carer_address AS ca
            WHERE ca.street_name = 'Hale End Road'
        );
    );
```



	first_name	last_name	class_name
	Evie	Jones	3AN
	Seb	Munroe	3AN

-- Example 2 for Subqueries --

```
SELECT  
    COUNT(p.id),  
    c.class_name  
FROM  
    pupil AS p  
    LEFT JOIN class AS c  
        ON c.id = p.class_id  
WHERE  
    p.id IN (  
        SELECT  
            a.pupil_id  
        FROM  
            attainment AS a  
        WHERE  
            a.term_id = 8  
        AND  
            a.attainment_level_id = 3  
        AND  
            a.subject_id = 2)  
GROUP BY 2;
```

How many children from each class are 'working towards' in their writing by the end of the Spring Term 2022?



	COUNT(p.id)	class_name
▶	6	3AN
	1	1LB

-- In your database, create a stored procedure and demonstrate how it runs --

DELIMITER //

CREATE PROCEDURE InsertNewPupil

```
IN id INT,  
IN PupilFirstName VARCHAR(50),  
IN PupilLastName VARCHAR(50),  
IN BirthDate DATE,  
IN Gender ENUM ('Male', 'Female', 'Other'),  
IN DateOfAdmission DATE,  
IN ClassID INT
```

```
)  
BEGIN  
    INSERT INTO pupil (id, first_name,  
                      last_name, date_of_birth, gender,  
                      admission_date, class_id)
```

VALUES (id, PupilFirstName, PupilLastName, BirthDate, Gender, DateOfAdmission, ClassID);

END //

DELIMITER ;

Entering new pupils details...



-- In your database, create a stored procedure and demonstrate how it runs --

CALL InsertNewPupil, (11, 'Ash', 'Ketchum', '2018-11-10', 'Male', '2023-05-28', 4)

```
SELECT *  
FROM pupil;
```



-- In your database, create a trigger and demonstrate how it runs --

Ensuring all pupil names are entered in the correct format...

DELIMITER //

```
CREATE TRIGGER name_before_insert
BEFORE INSERT ON pupil
FOR EACH ROW
BEGIN
    SET NEW.first_name = CONCAT(UPPER(LEFT(NEW.first_name, 1)),
                                LOWER(SUBSTRING(NEW.first_name FROM 2)));
    SET NEW.last_name = CONCAT(UPPER(LEFT(NEW.last_name, 1)),
                               LOWER(SUBSTRING(NEW.last_name FROM 2)));
END //
```

DELIMITER ;

```
CALL InsertNewPupil (12, 'beth', 'barrow', '2014-01-21', 'Female', '2023-05-28', 1);
```



-- In your database, create a trigger and demonstrate how it runs --

SELECT *
FROM pupil;



	id	first_name	last_name	date_of_birth	gender	admission_date	class_id
▶	1	Robin	Dale	2014-10-05	Male	2018-09-01	4
	2	Evie	Jones	2015-01-17	Female	2018-09-01	4
	3	Abrar	Ali	2015-01-31	Male	2018-09-01	4
	4	Jessie	Jackson	2015-04-04	Male	2018-09-01	4
	5	Seb	Munroe	2014-11-11	Male	2018-09-01	4
	6	Amelia	Sabbit	2014-12-05	Female	2018-12-05	4
	7	Ruby	Greenup	2017-05-02	Female	2021-09-01	2
	8	Kay	Arceus	2017-07-04	Male	2021-09-01	2
	9	Ruby	Arceus	2017-07-04	Female	2021-09-01	2
	10	Phoebe	Peters	2017-07-11	Female	2021-09-01	2
	12	Beth	Barrow	2014-01-21	Female	2023-05-28	1
	14	Zak	Greenup	2015-05-19	Male	2022-12-04	4
		NULl	NULl	NULl	NULl	NULl	NULl

-- In your database, create an event and demonstrate how it runs --

Monitoring updates on the pupil table:



```
CREATE TABLE monitoring_pupil (
    id INT NOT NULL AUTO_INCREMENT,
    Last_Update TIMESTAMP,
    PRIMARY KEY (id));
```

```
SELECT *
FROM monitoring_pupil
ORDER BY id DESC;
```

The screenshot shows the MySQL Workbench interface with a result grid. The grid has two columns: 'id' and 'Last_Update'. There are no data rows present in the grid.

	id	Last_Update

-- In your database, create an event and demonstrate how it runs --

DELIMITER //

```
CREATE EVENT recurring_event_monitoring_pupil
ON SCHEDULE EVERY 1 MINUTE
STARTS NOW()
DO BEGIN
    INSERT INTO monitoring_pupil(Last_Update)
    VALUES (NOW());
END//
```

DELIMITER ;

```
SELECT *
FROM monitoring_pupil
ORDER BY id DESC;
```



After 2 minutes...

<i>id</i>	<i>Last_Update</i>
2	2023-05-30 21:23:24
1	2023-05-30 21:22:24

-- Create a view that uses at least 3-4 base tables --

VIEW 1:

```
CREATE VIEW vw_pupil_class
AS
SELECT
    p.id,
    p.first_name,
    p.last_name,
    p.class_id,
    c.class_name
FROM pupil AS p
LEFT JOIN class AS c
    ON c.id = p.class_id
ORDER BY class DESC;

SELECT *
FROM vw_pupil_class;
```

id	first_name	last_name	class_id	class_name
1	Robin	Dale	4	3AN
2	Evie	Jones	4	3AN
3	Abrar	Ali	4	3AN
4	Jessie	Jackson	4	3AN
5	Seb	Munroe	4	3AN
6	Amelia	Sabbit	4	3AN
7	Ruby	Greenup	2	1LB
8	Kay	Arceus	2	1LB
9	Ruby	Arceus	2	1LB
10	Phoebe	Peters	2	1LB
14	Zak	Greenup	4	3AN

-- Create a view that uses at least 3-4 base tables --

VIEW 2:



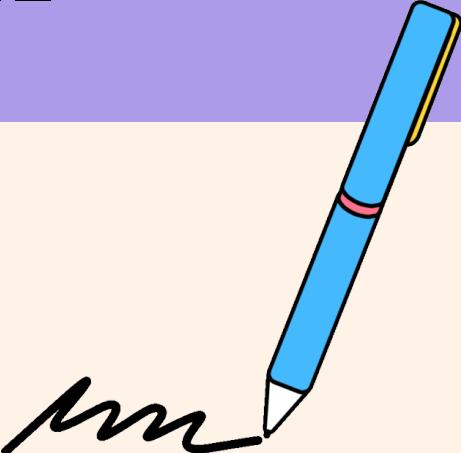
```
CREATE VIEW vw_writing_attainment_Autumn22
AS
SELECT
    pupil_id,
    attainment_level_id AS Autumn_2022
FROM attainment
WHERE term_id = 7
    AND subject_id = 2;

SELECT *
FROM vw_writing_attainment_Autumn22;
```

pupil_id	Autumn_2022
1	1
2	3
3	2
4	3
5	2
6	3
7	2
8	2
9	2
10	1
14	1

-- Create a view that uses at least 3-4 base tables --

VIEW 3:



```
CREATE VIEW vw_writing_attainment_Spring22
AS
SELECT
    pupil_id,
    attainment_level_id AS Spring_2022
FROM attainment
WHERE term_id = 8
    AND subject_id = 2;

SELECT *
FROM vw_writing_attainment_Spring22;
```

pupil_id	Spring_2022
1	3
2	3
3	2
4	3
5	3
6	3
7	2
8	3
9	2
10	1
14	3

-- Create a view that uses at least 3-4 base tables --

VIEW 4:

```
CREATE VIEW vw_class_send
AS
SELECT
    sr.pupil_id AS pupil_id,
    sr.send_id AS send_id,
    s.category AS send
FROM send_register AS sr
LEFT JOIN send AS s
    ON s.id = sr.send_id;
SELECT *
FROM vw_class_send;
```

pupil_id	send_id	send
2	7	ASD
5	11	PD
8	7	ASD
9	5	Speech Language Communication Needs

-- Create a view that uses at least 3-4 base tables --

VIEW 5:

```
CREATE VIEW vw_intervention_groups
AS
SELECT
    i.pupil_id,
    it.intervention_name
FROM intervention AS i
LEFT JOIN intervention_type AS it
    ON it.id = i.intervention_type_id;

SELECT *
FROM vw_intervention_groups;
```

pupil_id	intervention_name
1	play_therapy
1	social_worker
4	writing_group
6	phonics
6	writing_group
8	SLT
8	phonics
8	maths_group
8	play_therapy
5	SLT
9	SLT
9	phonics
9	writing_group
2	one-to-one

-- Create a view that uses at least
3-4 base tables --

```
CREATE VIEW vw_writing_attainment2022
AS
SELECT
    p.first_name,
    p.last_name,
    p.class_name,
    aw.Autumn_2022,
    sw.Spring_2022,
    cs.send,
    ig.intervention_name
FROM vw_pupil_class AS p
JOIN vw_writing_attainment_Autumn22 AS aw
    ON aw.pupil_id = p.id
JOIN vw_writing_attainment_Spring22 AS sw
    ON sw.pupil_id = p.id
LEFT JOIN vw_class_send AS cs
    ON cs.pupil_id = p.id
LEFT JOIN intervention_groups AS ig
    ON ig.pupil_id = p.id
GROUP BY 3, 1, 2, 4, 5, 6, 7
ORDER BY p.class_name DESC;
```



```
SELECT *
FROM vw_writing_attainment2022;
```

-- Create a view that uses at least 3-4 base tables --

first_name	last_name	class_name	Autumn_2022	Spring_2022	send	intervention_name
Abrar	Ali	3AN	2	2	NUL	NUL
Amelia	Sabbit	3AN	3	3	NUL	phonics
Amelia	Sabbit	3AN	3	3	NUL	writing_group
Evie	Jones	3AN	3	3	ASD	one-to-one
Jessie	Jackson	3AN	3	3	NUL	writing_group
Robin	Dale	3AN	1	3	NUL	play_therapy
Robin	Dale	3AN	1	3	NUL	social_worker
Seb	Munroe	3AN	2	3	PD	SLT
Zak	Greenup	3AN	1	3	NUL	NUL
Kay	Arceus	1LB	2	3	ASD	maths_group
Kay	Arceus	1LB	2	3	ASD	phonics
Kay	Arceus	1LB	2	3	ASD	play_therapy
Kay	Arceus	1LB	2	3	ASD	SLT
Phoebe	Peters	1LB	1	1	NUL	NUL
Ruby	Arceus	1LB	2	2	Speech Language Communication Needs	phonics
Ruby	Arceus	1LB	2	2	Speech Language Communication Needs	SLT
Ruby	Arceus	1LB	2	2	Speech Language Communication Needs	writing_group
Ruby	Greenup	1LB	2	2	NUL	NUL

-- prepare and demonstrate a query that uses the view to produce a logically arranged result set for analysis --

Who is 'Working Towards' in their writing by the end of the Spring Term 2022 in "3AN"? Why are they 'Working Towards'? Do they have any interventions in place to show that we're supporting them in their writing?

SELECT

```
first_name,  
last_name,  
send,  
intervention_name  
FROM vw_writing_attainment2022  
WHERE Spring_2022 = 3  
AND class_name = '3AN';
```

Zak Greenup is working towards in writing but is not in any support groups - plan some intervention to support Zak.

first_name	last_name	send	intervention_name
Robin	Dale	NULL	play_therapy
Robin	Dale	NULL	social_worker
Evie	Jones	ASD	one-to-one
Jessie	Jackson	NULL	writing_group
Seb	Munroe	PD	SLT
Amelia	Sabbit	NULL	phonics
Amelia	Sabbit	NULL	writing_group
Zak	Greenup	NULL	NULL

-- Example 2 --

Who is 'Working Towards' in their writing by the end of the Spring Term 2022 in "1LB"? Why are they 'Working Towards'? Do they have any interventions in place to show that we're supporting them in their writing?

```
SELECT  
    first_name,  
    last_name,  
    send,  
    intervention_name  
FROM vw_writing_attainment2022  
WHERE Spring_2022 = 3  
AND class_name = '1LB';
```

first_name	last_name	send	intervention_name
Kay	Arceus	ASD	SLT
Kay	Arceus	ASD	phonics
Kay	Arceus	ASD	maths_group
Kay	Arceus	ASD	play_therapy

Kay is still working towards in writing. She has Autism and is in many support groups. Review these support groups and monitor her progress for the next term. Review her learning passport and targets. Speak with her teacher as she may have made progress within this band.

-- Example 3 --

Who has fallen behind in their writing from Autumn 2022? Why? Do we have anything in place to support them?

```
SELECT  
    first_name,  
    last_name,  
    intervention_name,  
    send  
FROM vw_writing_attainment2022  
WHERE Autumn_2022 < 3  
AND Spring_2022 = 3;
```

Zak Greenup has fallen behind in writing but is not in any support group. Query why he may be falling behind and what support will be put in place moving forward.

first_name	last_name	intervention_name	send
Robin	Dale	play_therapy	HULL
Robin	Dale	social_worker	HULL
Seb	Munroe	SLT	PD
Zak	Greenup	HULL	HULL
Kay	Arceus	maths_group	ASD
Kay	Arceus	phonics	ASD
Kay	Arceus	play_therapy	ASD
Kay	Arceus	SLT	ASD

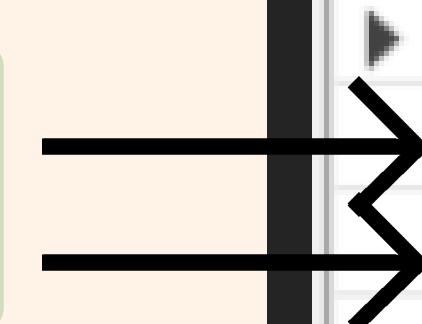
-- Example 4 --

Are children on our SEND register making expected progress in writing?

```
SELECT  
first_name,  
last_name,  
Autumn_2022,  
Spring_2022,  
class_name  
FROM vw_writing_attainment2022  
WHERE send != 'NULL'  
GROUP BY 5, 1, 2, 3, 4;
```

Both have fallen behind in their writing! Find out why and what further support can be put in place.

Review their learning passports and targets.

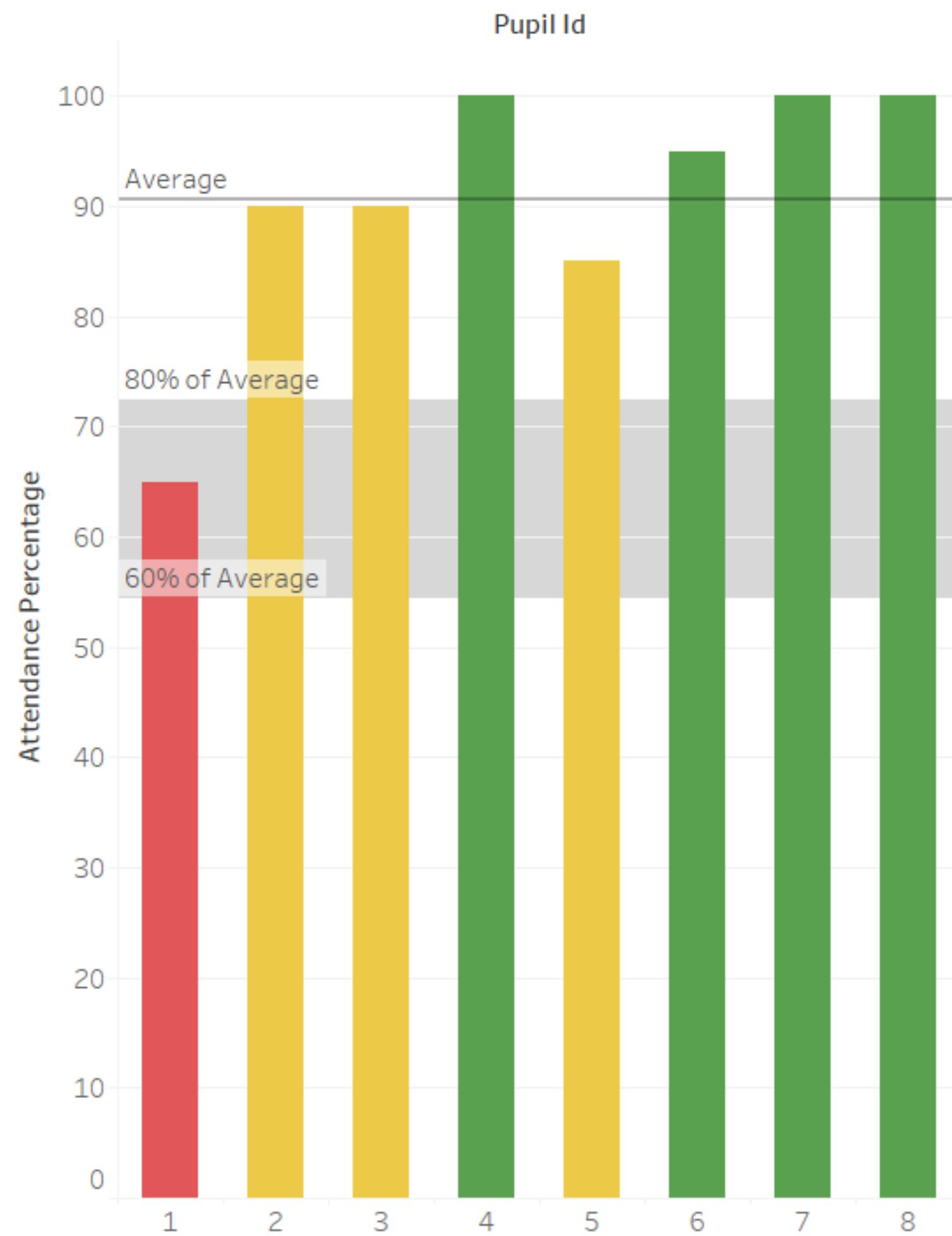


	first_name	last_name	Autumn_2022	Spring_2022	class_name
▶	Evie	Jones	3	3	3AN
▶	Seb	Munroe	2	3	3AN
▶	Kay	Arceus	2	3	1LB
▶	Ruby	Arceus	2	2	1LB

Pupil 1's attendance is around 70% below average. Talk to teacher to find out what may be going on. Are we aware of any additional support needed at school or at home? Check to see if his learning and welfare has been effected. Are there any previous safeguarding concerns? Talk to his carer/s.

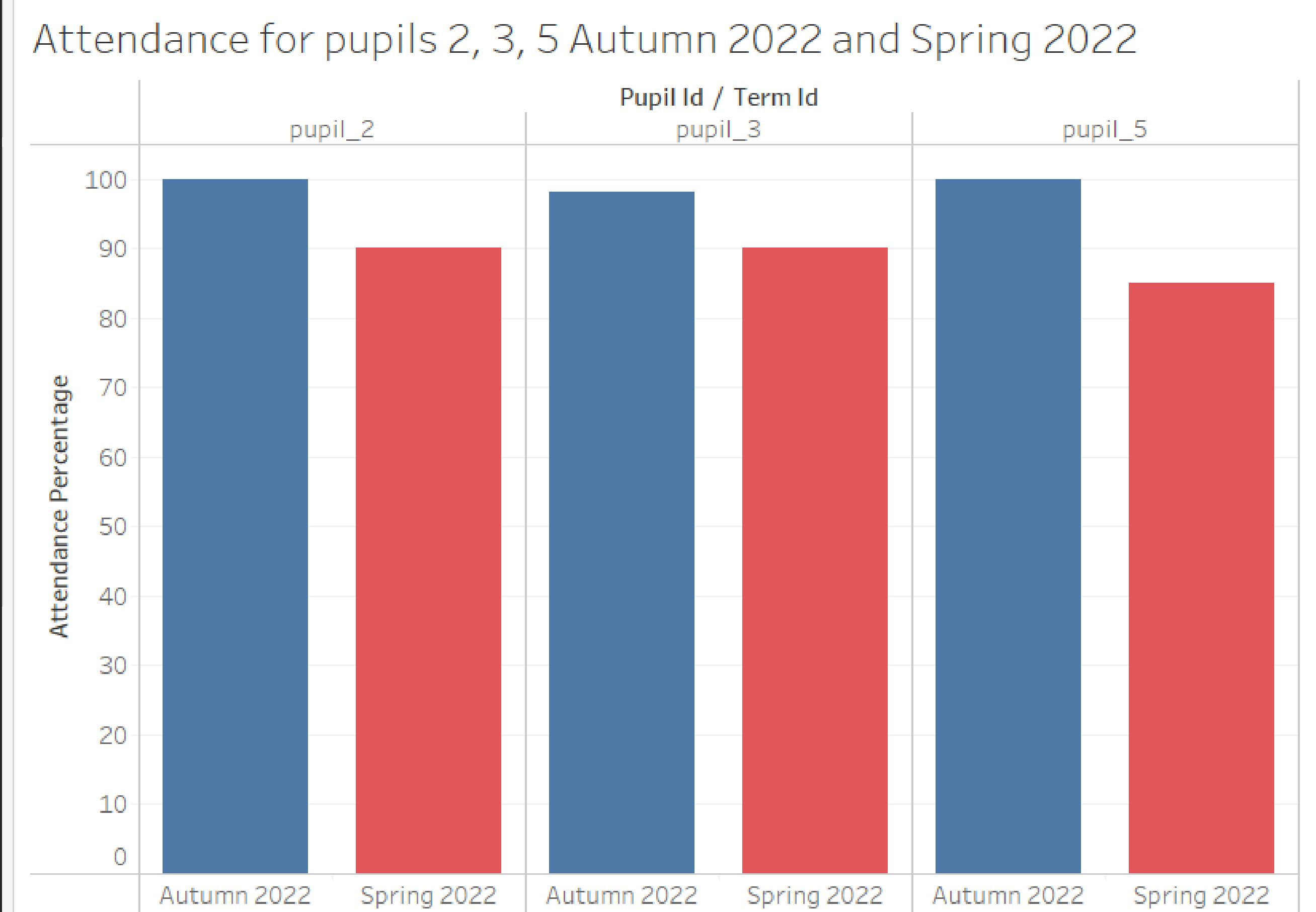
Pupil 2, 3, and 5 are below average in their attendance but above the 80% of average mark. Check their previous attendance records. See next slide...

Pupil Attendance

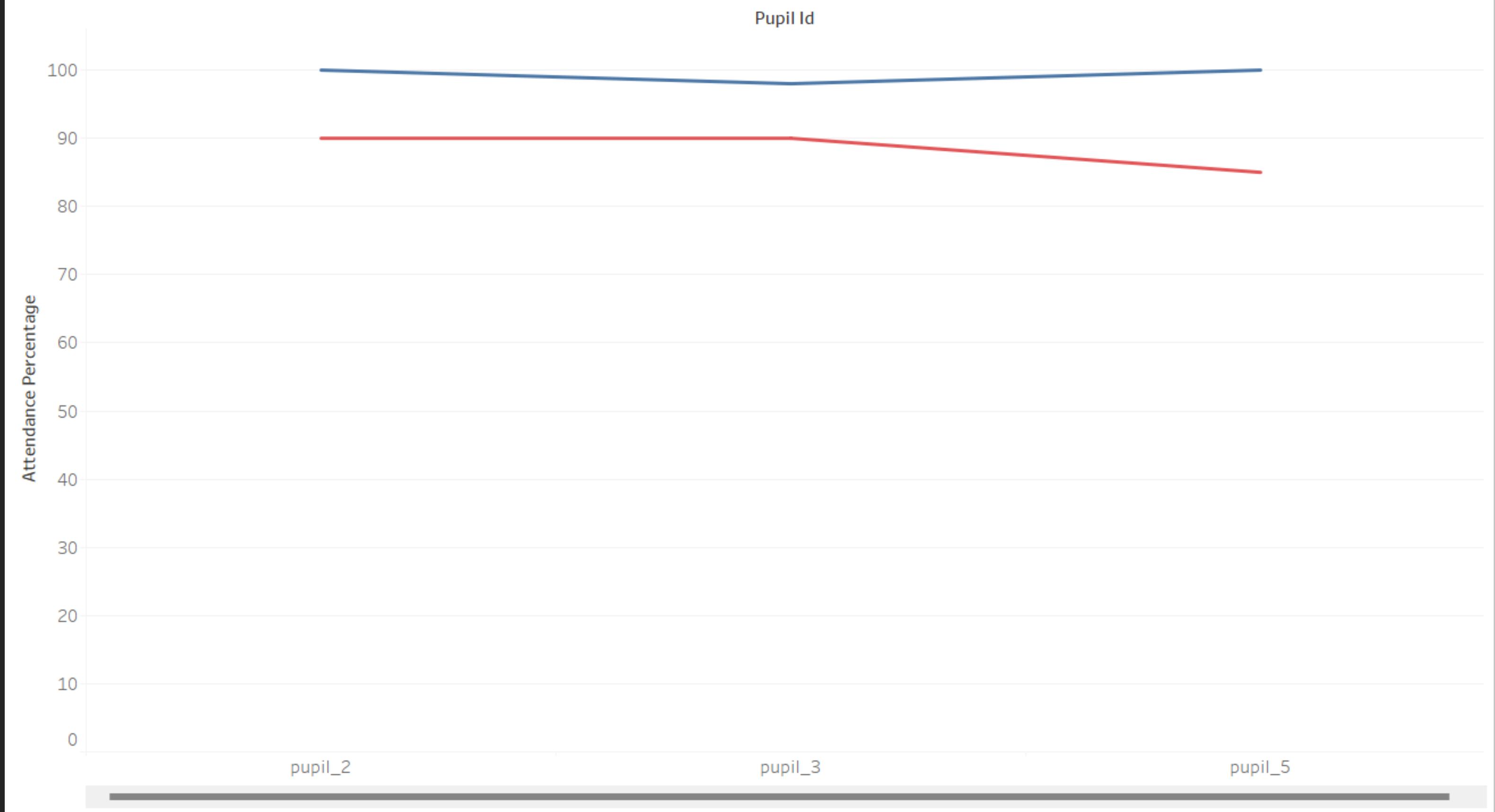


```
SELECT *  
FROM  
  attendance_percentage  
WHERE pupil_id  
  IN (2, 3, 5);
```

These pupils had good attendance in the Autumn term therefore the drop in attendance during the Spring term could be down to catching a sickness bug.
Monitor going forward.



Attendance for pupils 2, 3, 5 Autumn 2022 and Spring 2022



Term Id
Autumn 2022
Spring 2022



THANK YOU
FOR
YOUR TIME