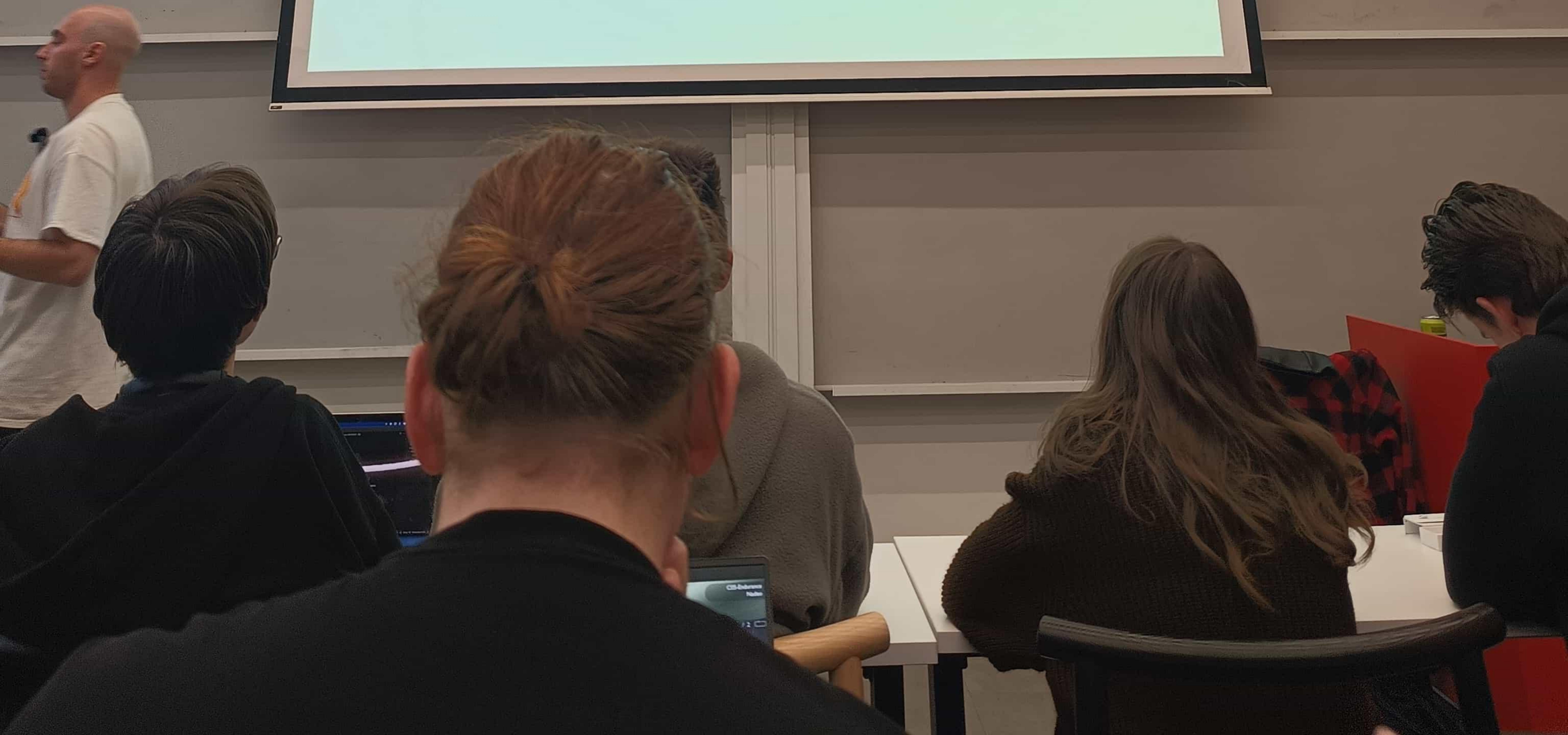


Реляционная модель

Атрибут

- У каждого атрибута должно быть уникальное имя в пределах своей таблицы
- В каждом кортеже содержится значение атрибута, которое принадлежит домену атрибута.



Реляционная модель

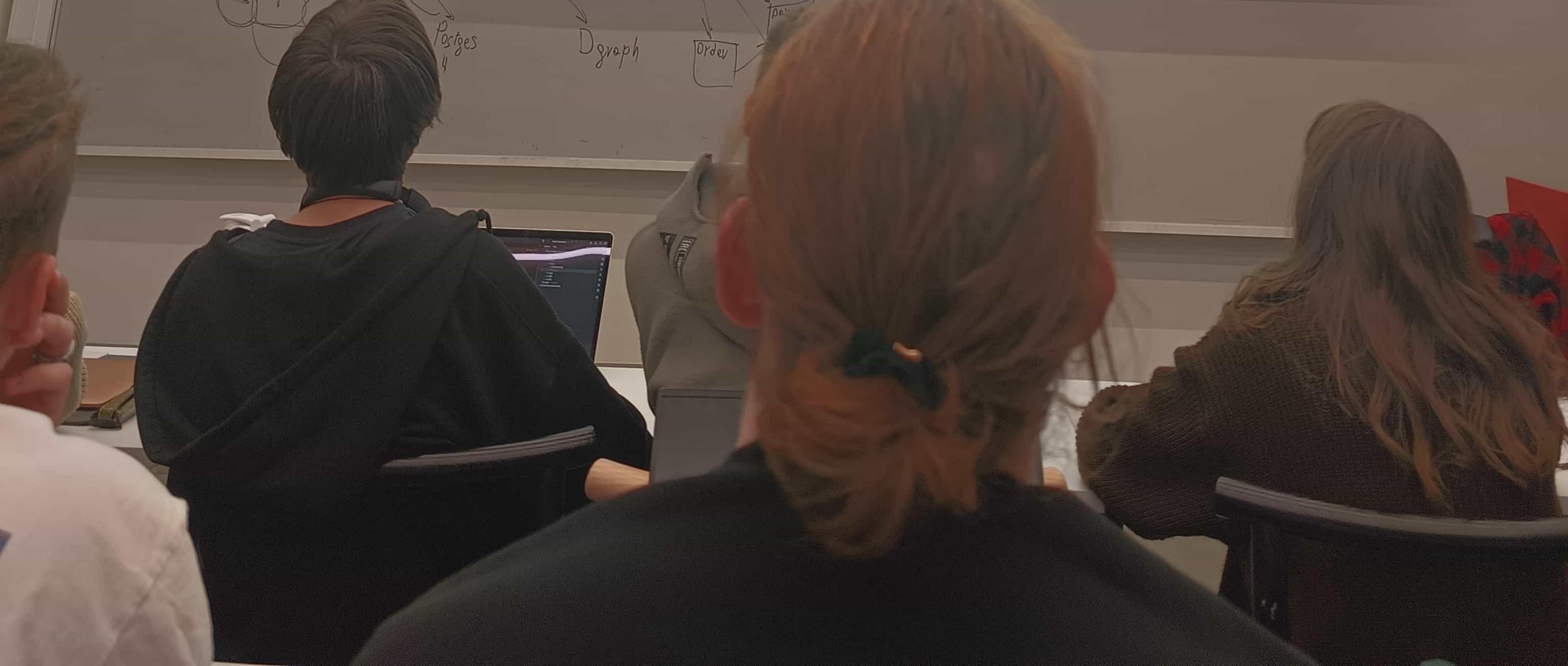
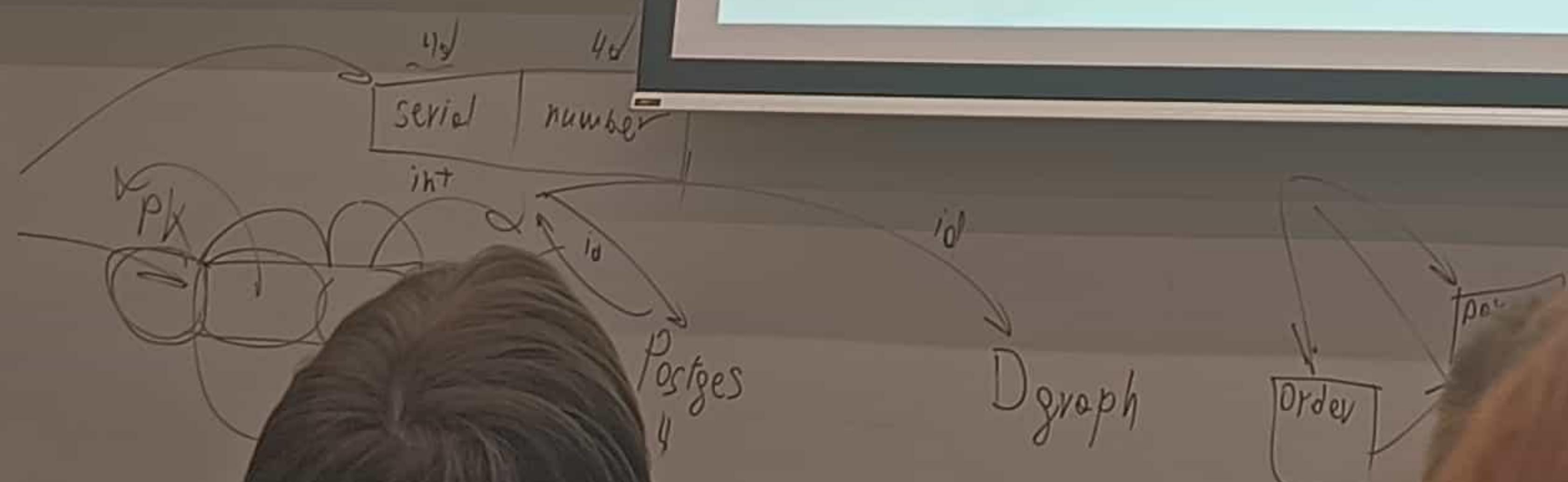
Виды атрибутов

- Простой
- Составной
- Однозначный
- Многозначный
- Производный

Реляционная модель

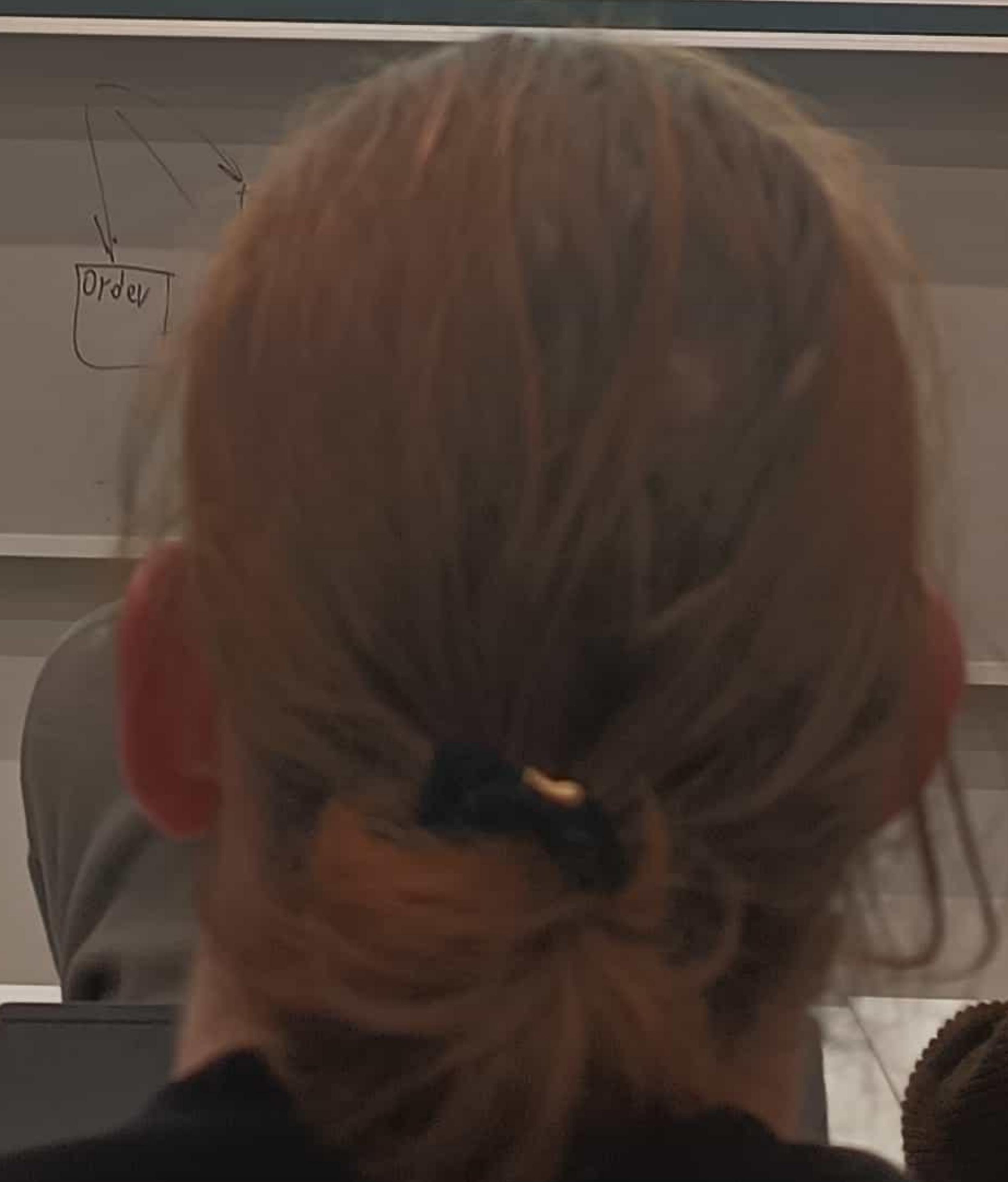
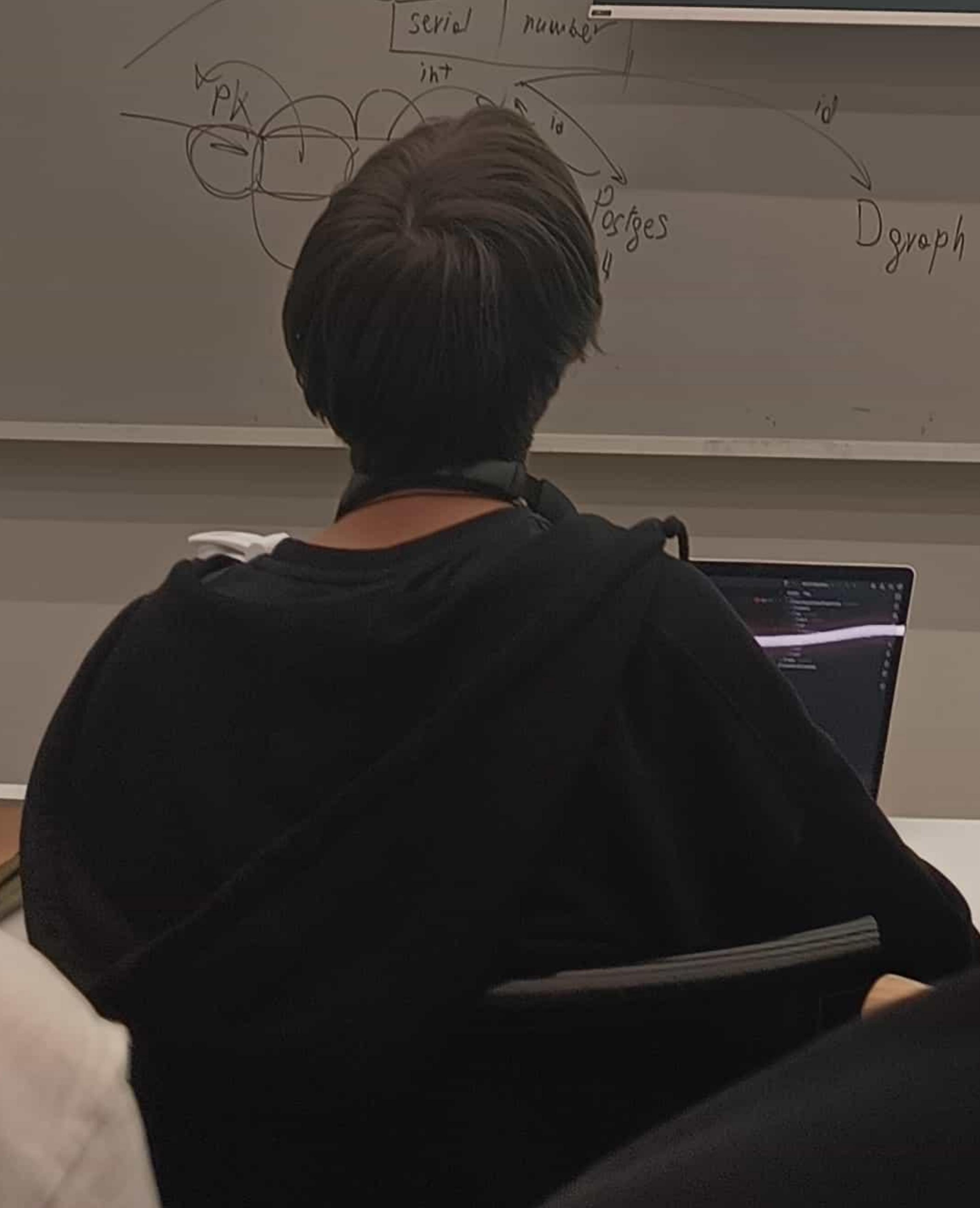
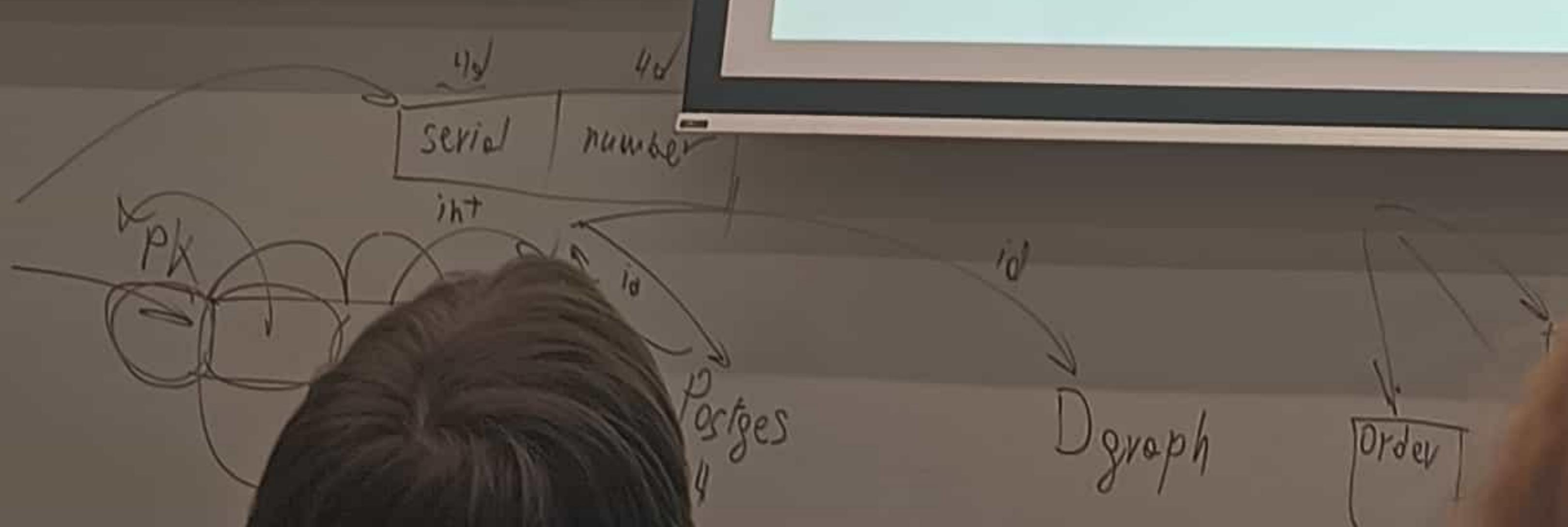
Типы данных

- Целочисленные типы (smallint, int, bigint)
- С плавающей точкой (real, double)
- Строки (varchar, char, text)
- Дата и время (date, time, timestamp, interval)
- Логический (bool)
- Геометрические типы (point, path, line)
- json



Реляционная модель Null

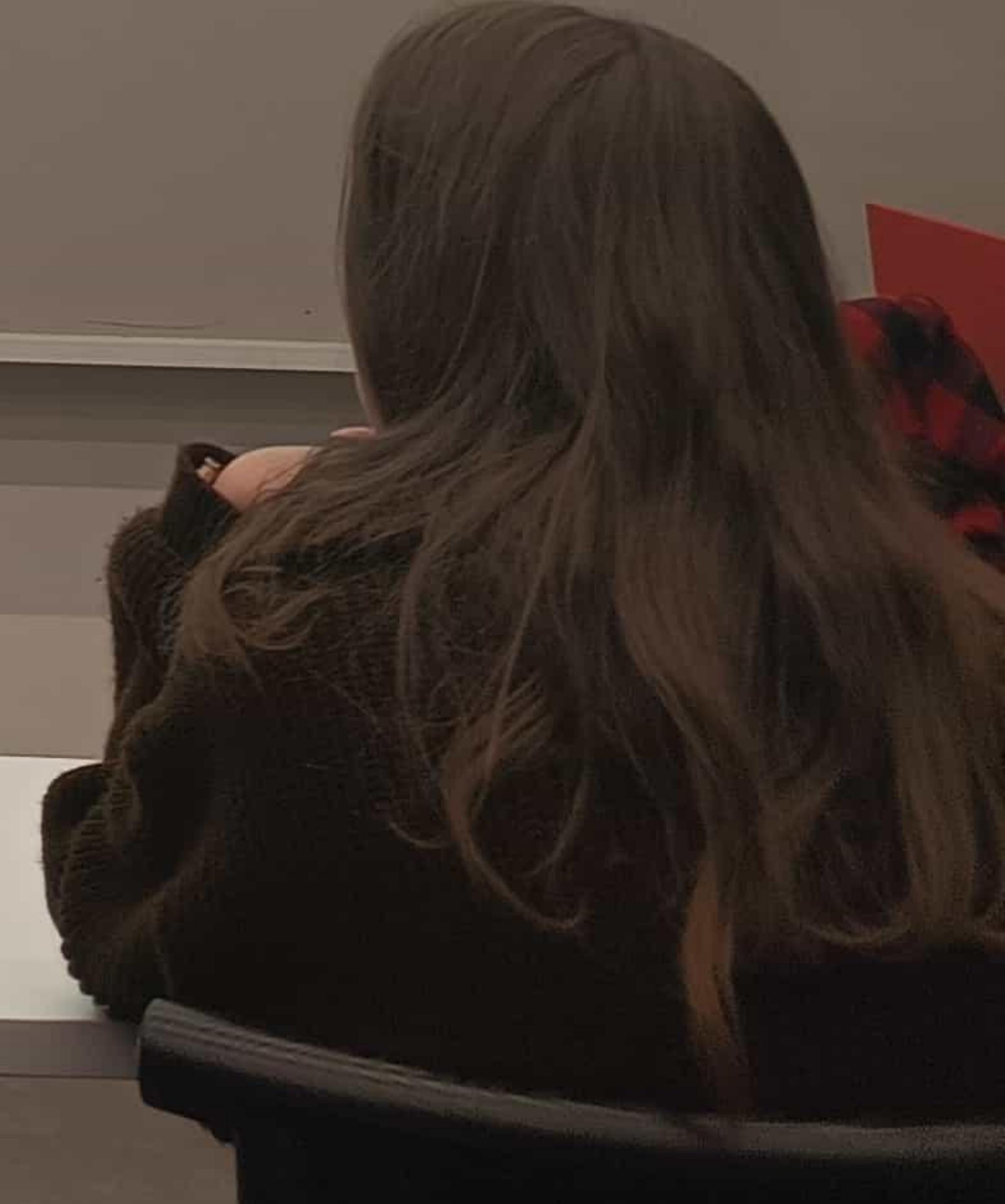
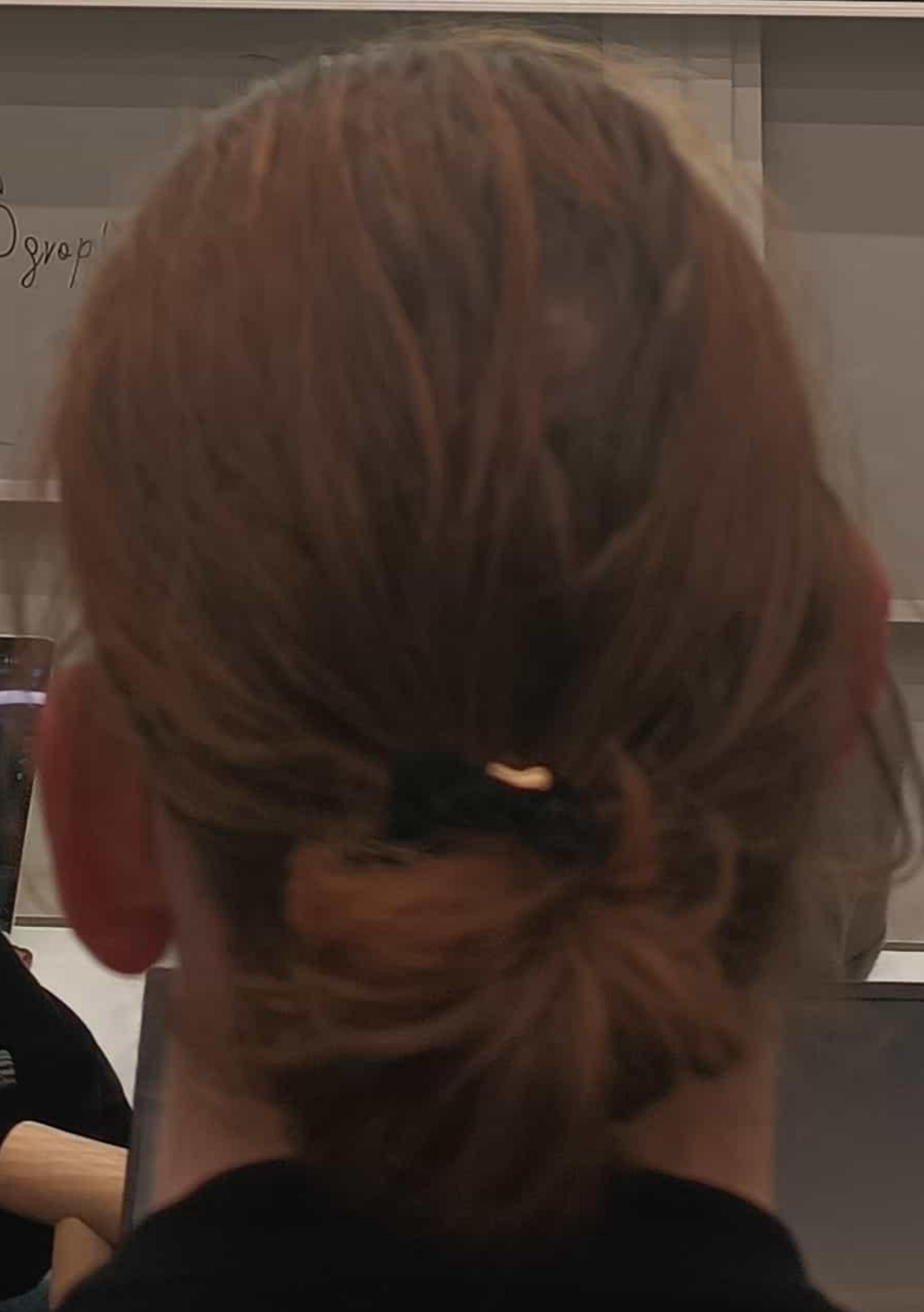
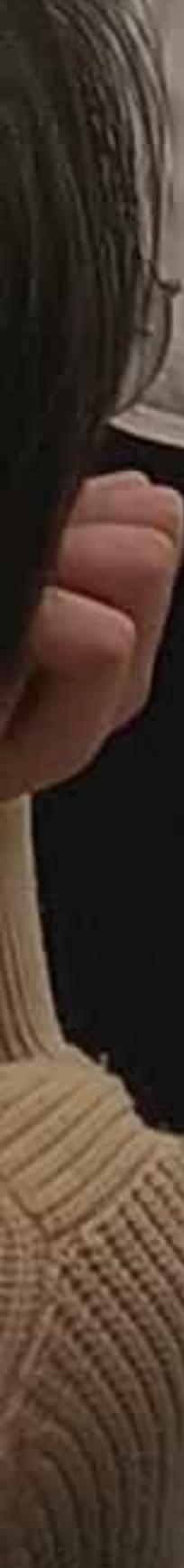
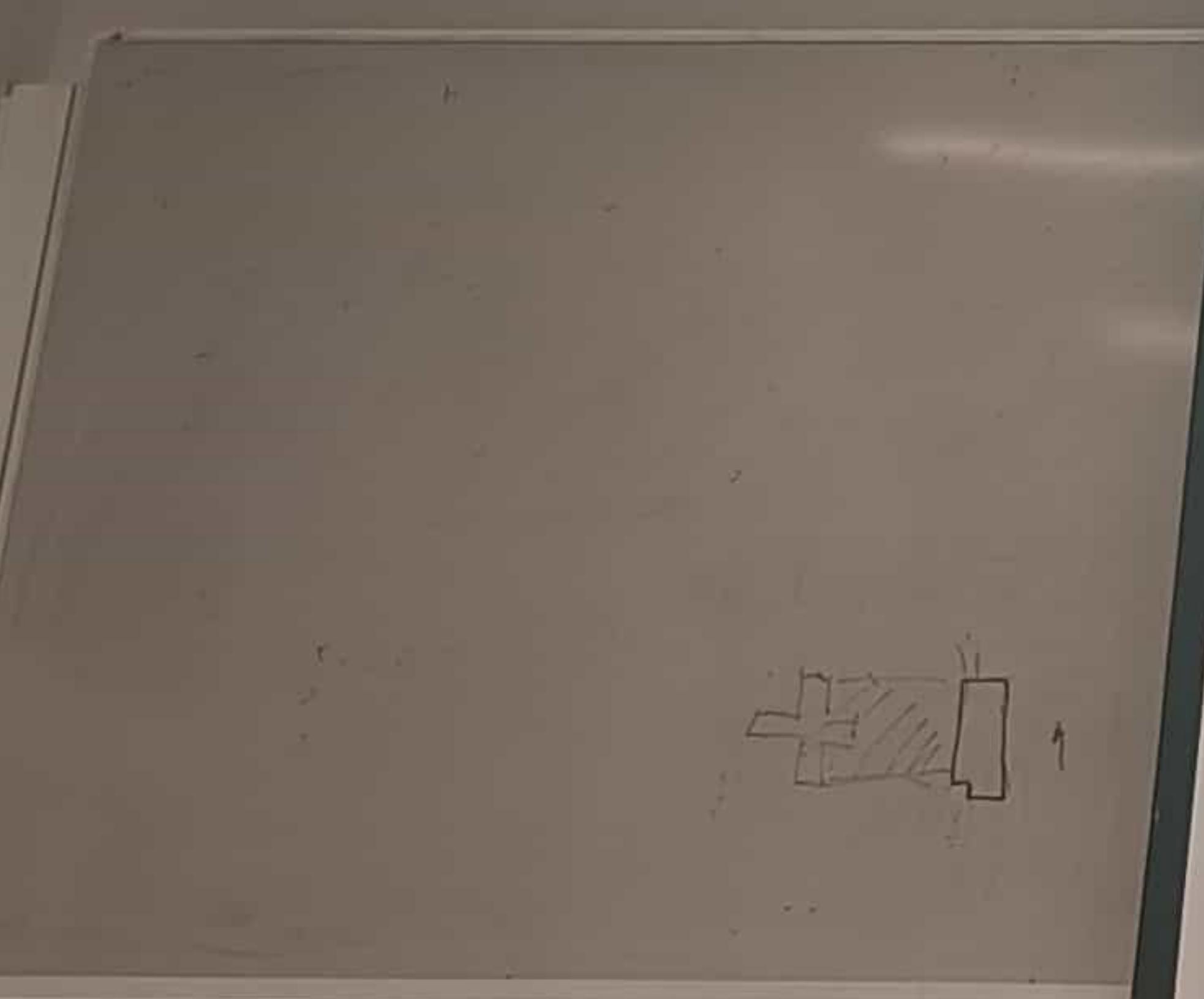
- $\text{SELECT } 1 = \text{NULL}$ вернет NULL
- $1 \diamond \text{NULL}$ вернет NULL
- $\text{SELECT } 1 \text{ IS NULL}$ вернет 0
- NULL IS NULL вернет 1



Реляционная модель

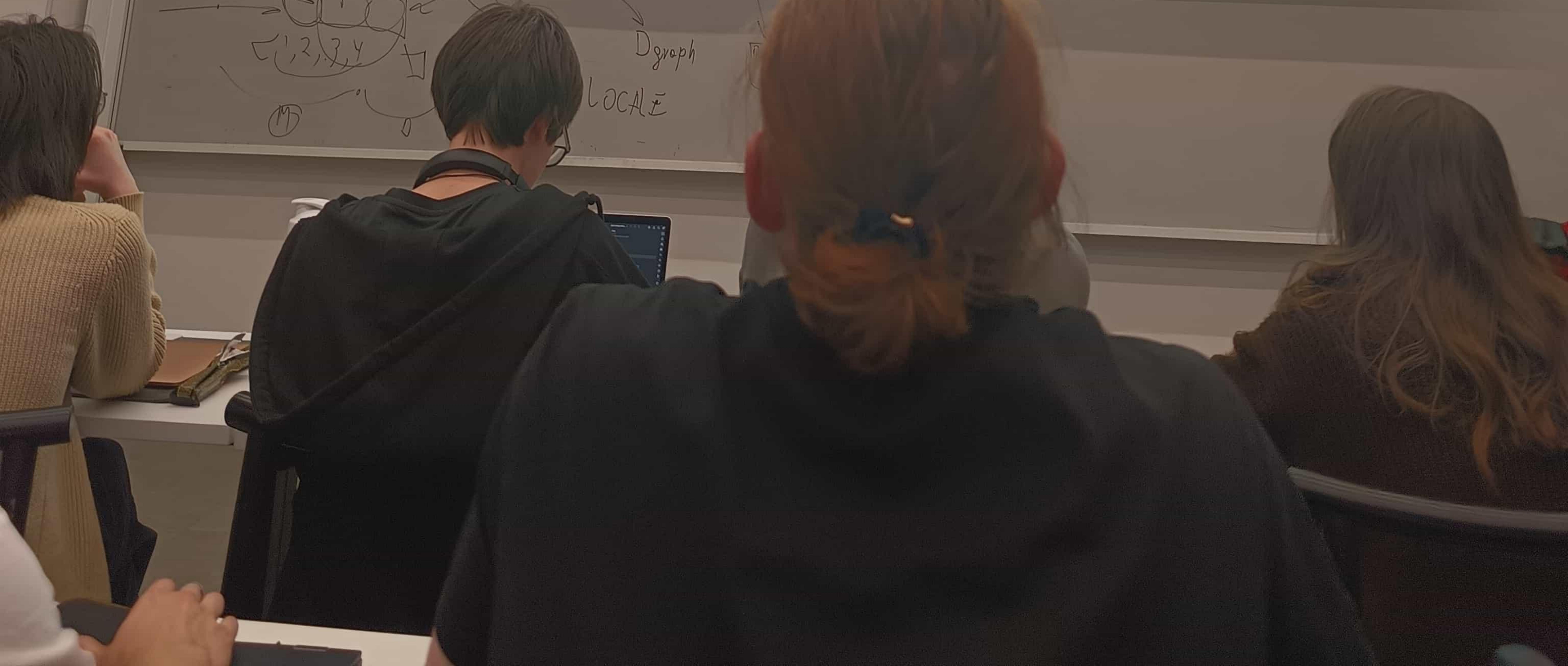
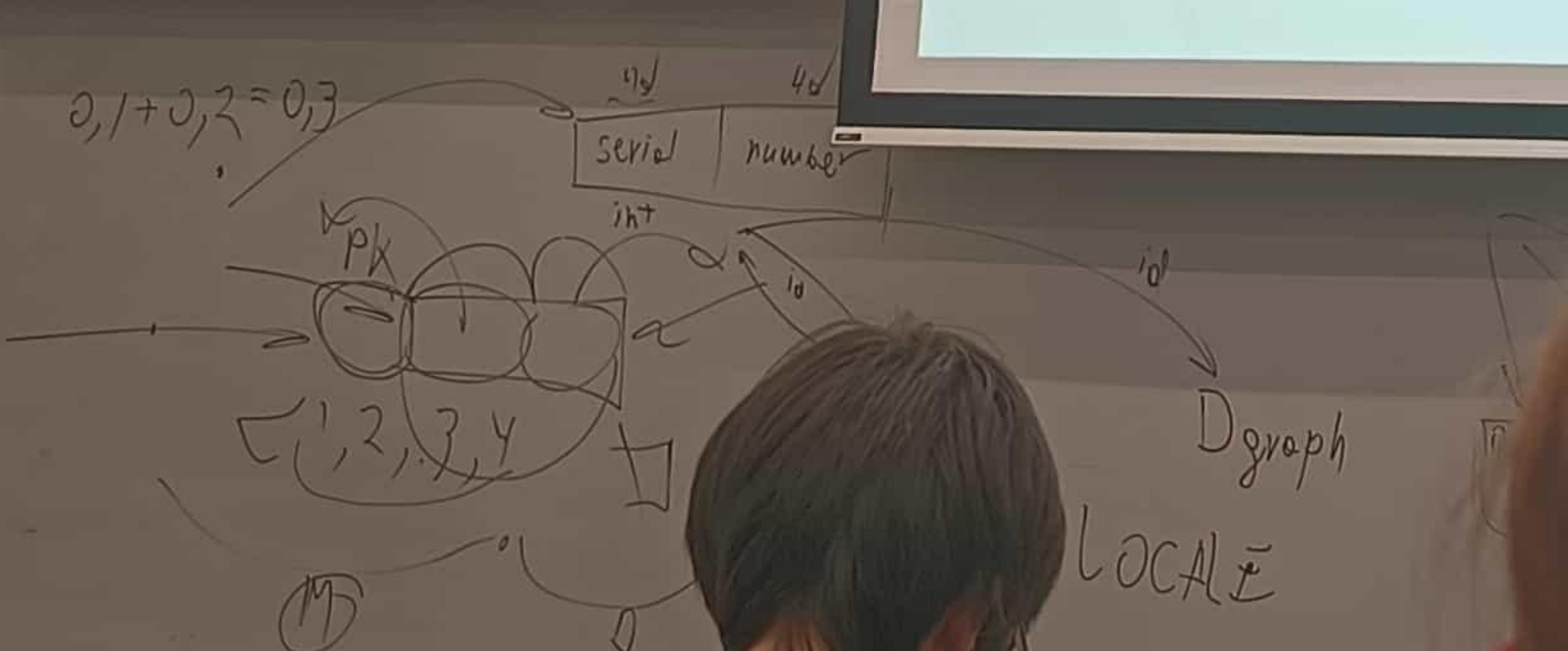
Числовые типы

- Фиксированные(SMALLINT, INT, BIGINT)
- DECIMAL(M, D)
- Плавающая точка (REAL, DOUBLE)
- Денежный
- Автотинкрементные(SMALLSERIAL, SERIAL, BIGSERIAL)



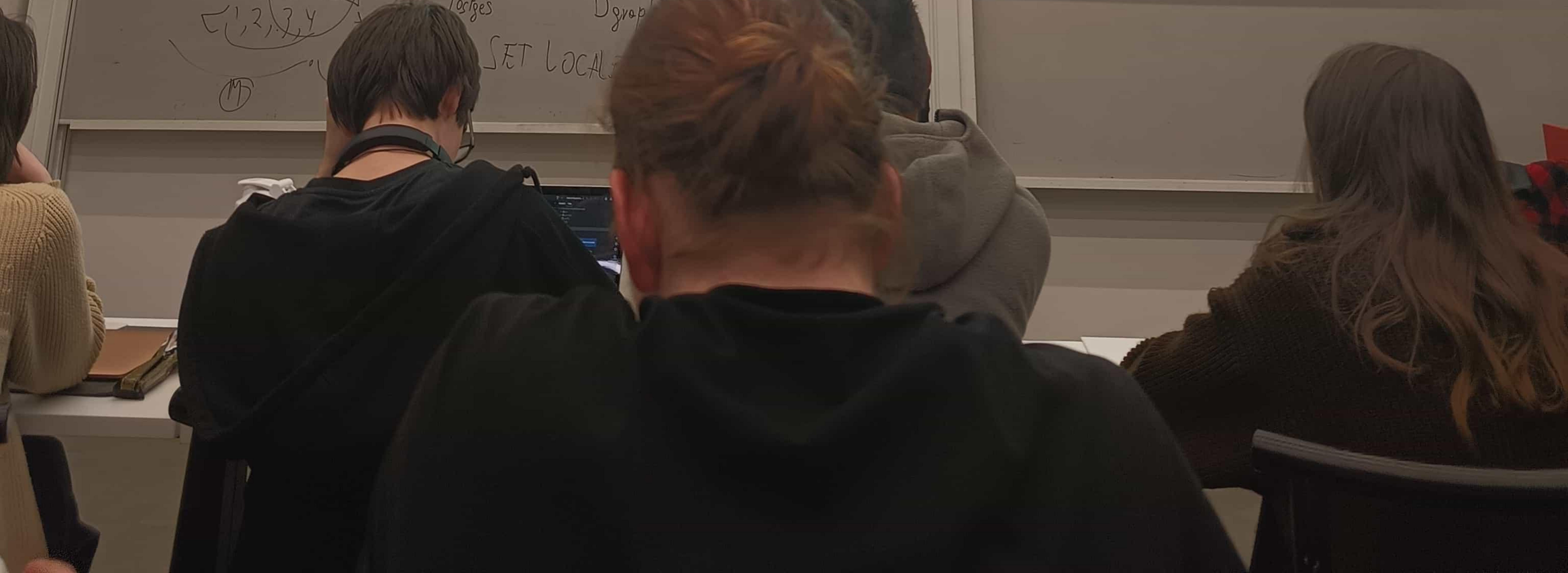
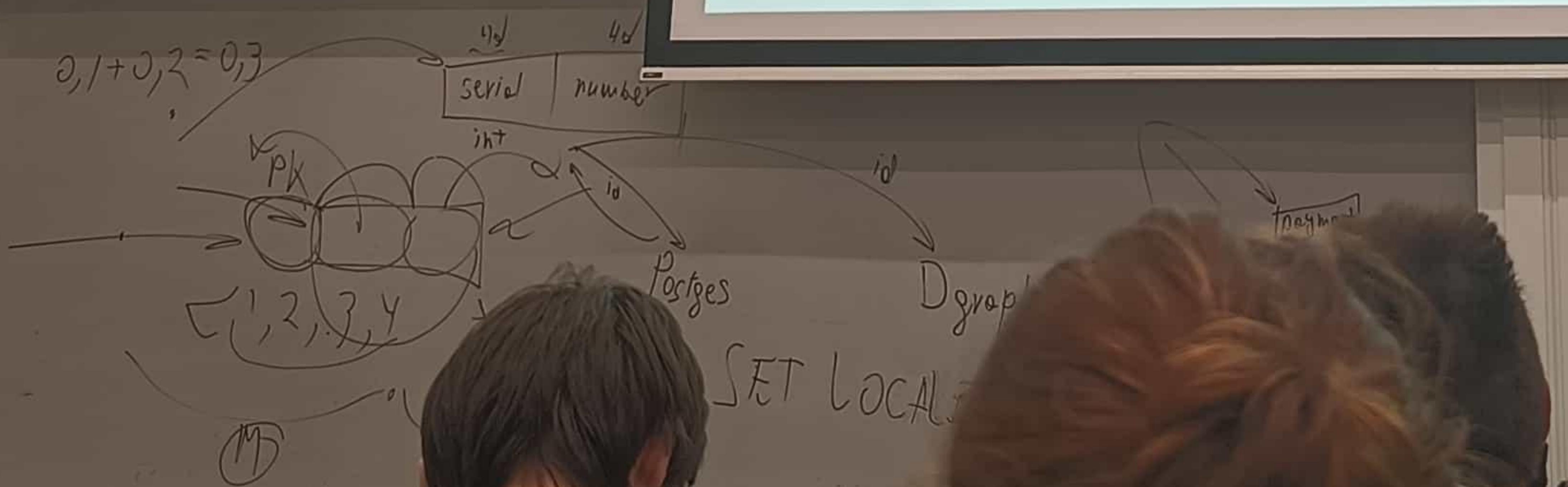
DML

- SELECT – выборка данных
- INSERT – вставка новых данных
- UPDATE – обновление данных
- DELETE – удаление данных
- MERGE – слияние данных



DDL

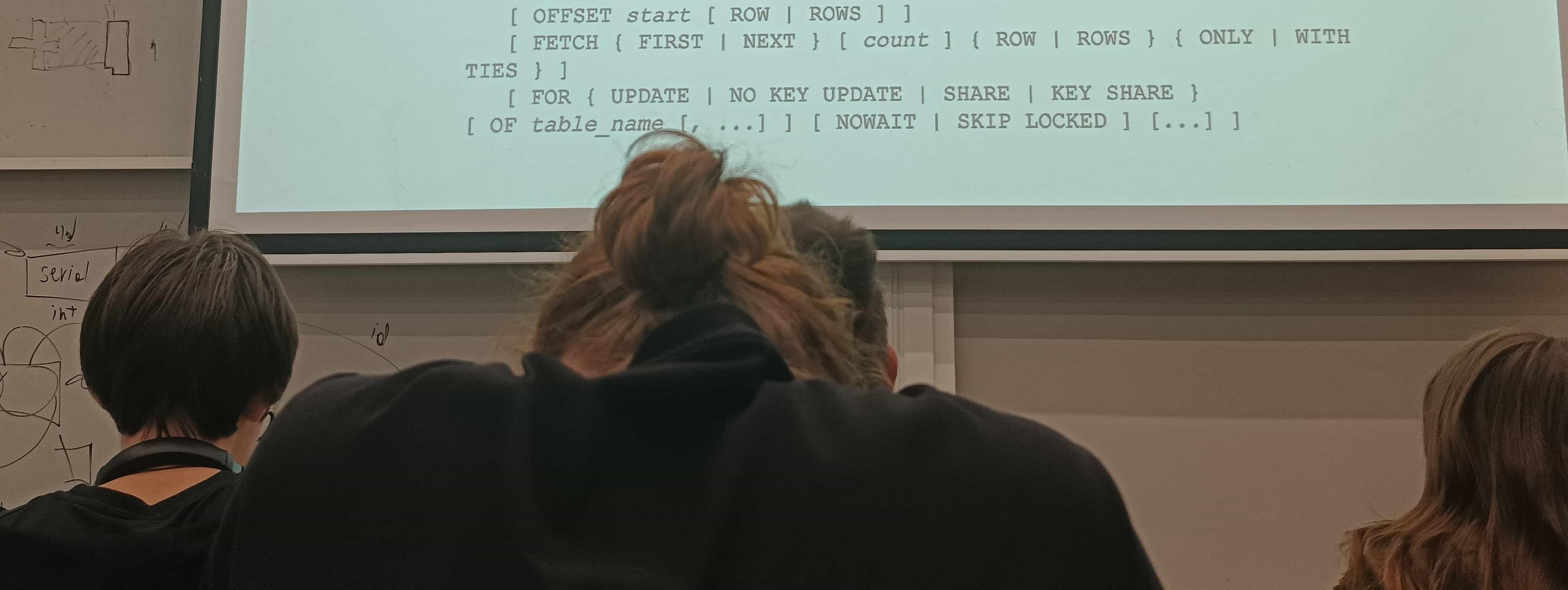
- CREATE TABLE
- ALTER TABLE
- DROP
- TRUNCATE



SELECT

Кусочек документации PostgreSQL 16

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
      [ * | expression [ [ AS ] output_name ] [, ...] ]
      [ FROM from_item [, ...] ]
      [ WHERE condition ]
      [ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
      [ HAVING condition ]
      [ WINDOW window_name AS ( window_definition ) [, ...] ]
      [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
      [ ORDER BY expression [ ASC | DESC | USING operator ] [ NULLS
        { FIRST | LAST } ] [, ...] ]
      [ LIMIT { count | ALL } ]
      [ OFFSET start [ ROW | ROWS ] ]
      [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } { ONLY | WITH
        TIES } ]
      [ FOR { UPDATE | NO KEY UPDATE | SHARE | KEY SHARE }
      [ OF table_name [, ...] ] [ NOWAIT | SKIP LOCKED ] [...] ]
```



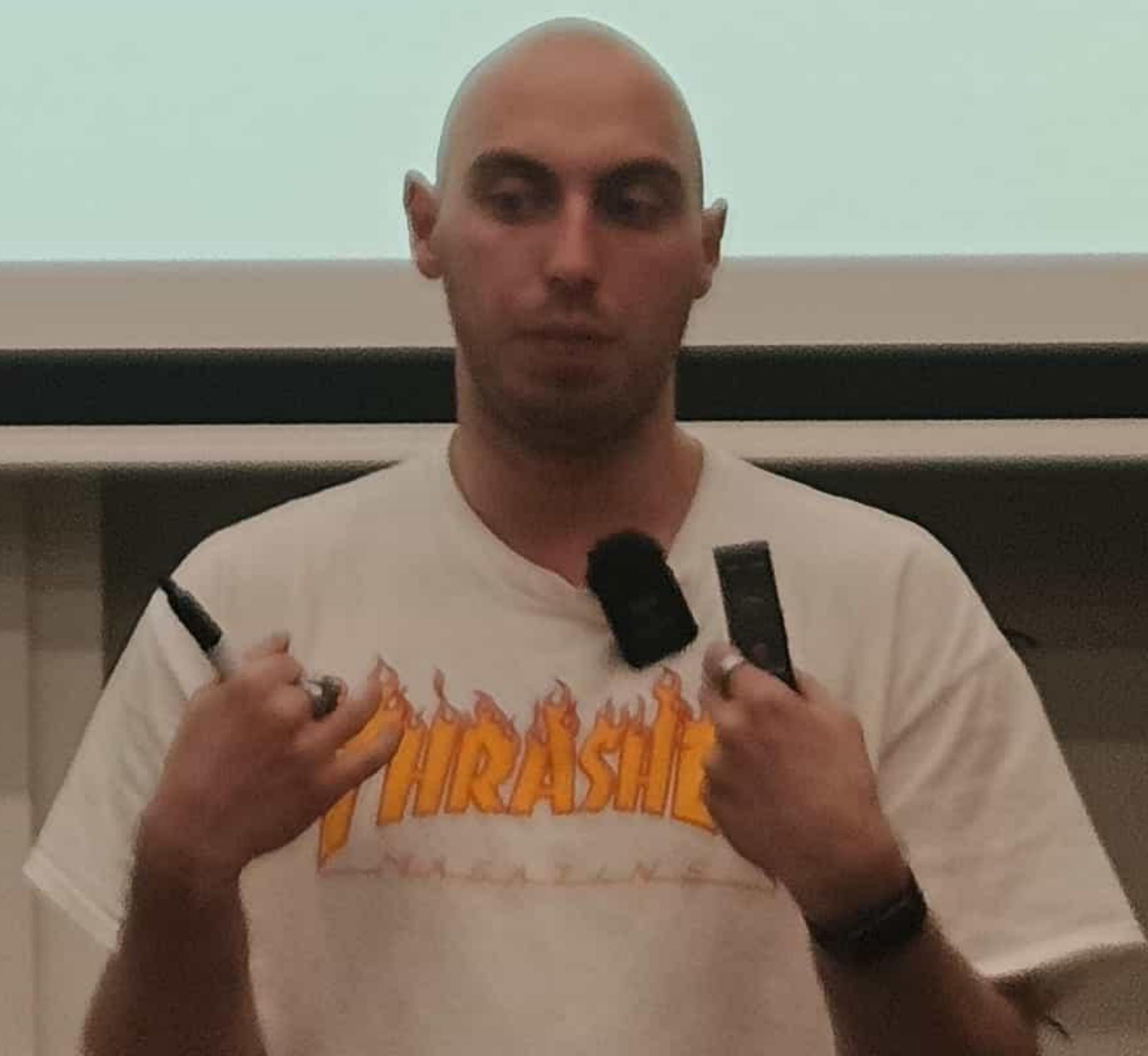
FROM

Выбор таблицы

```
SELECT *  
FROM SCHEMA.TABLE
```

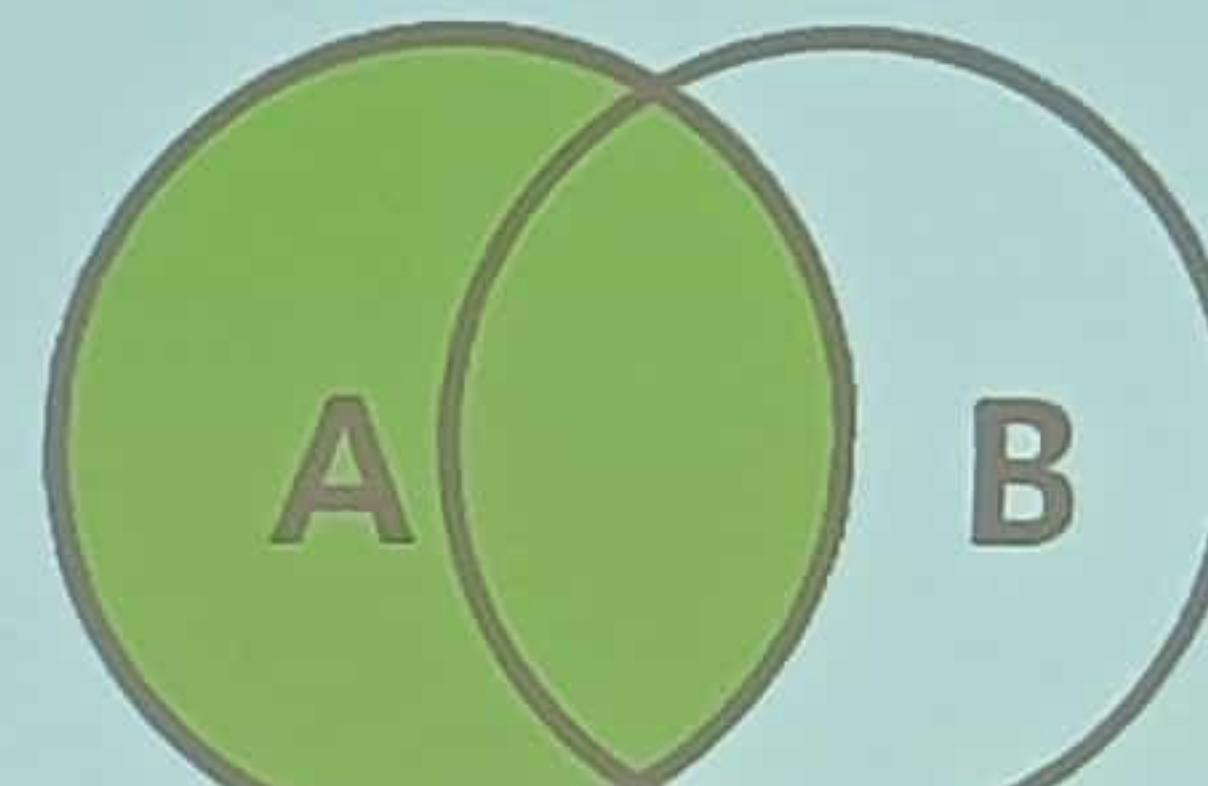
```
SELECT {a1, a2}  
FROM SCHEMA.TABLE
```

```
SELECT * FROM (SELECT id, name FROM employees) AS emp_sub;
```

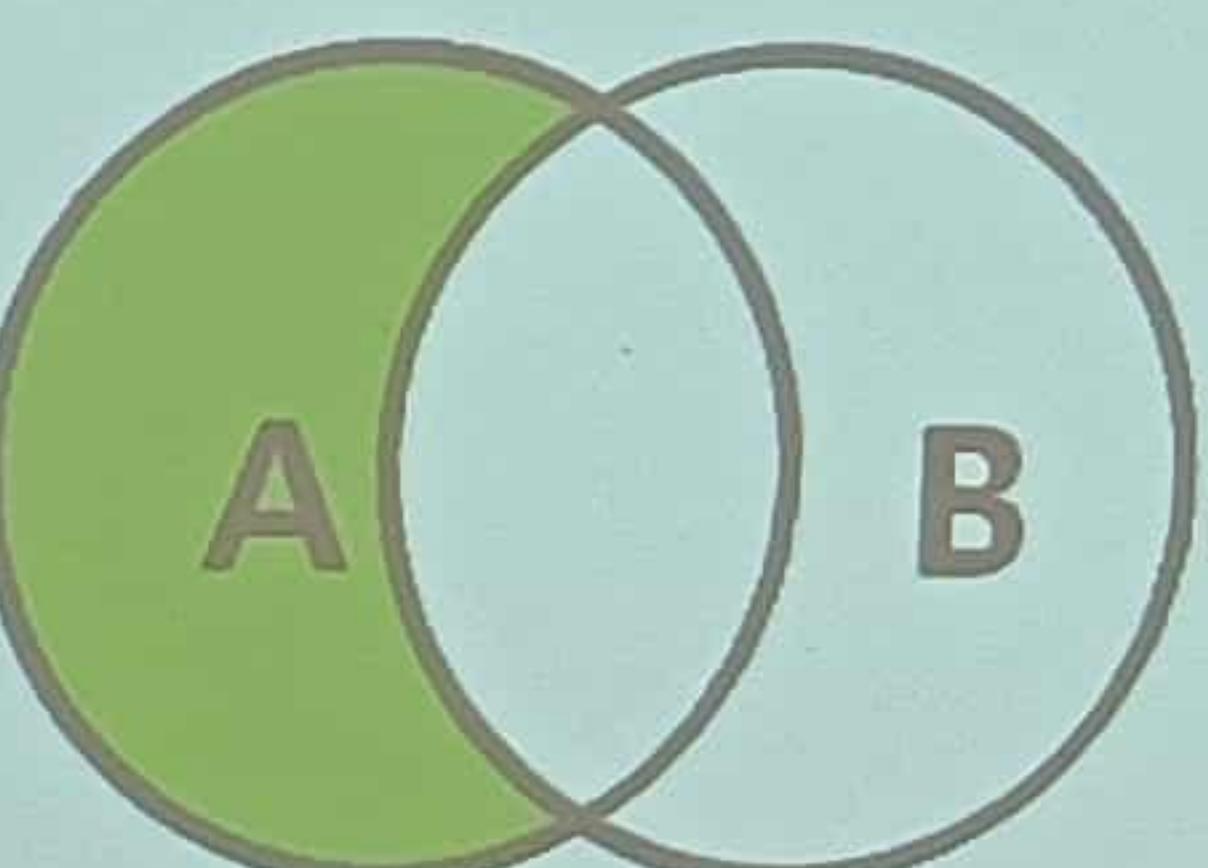


JOIN

Объединение

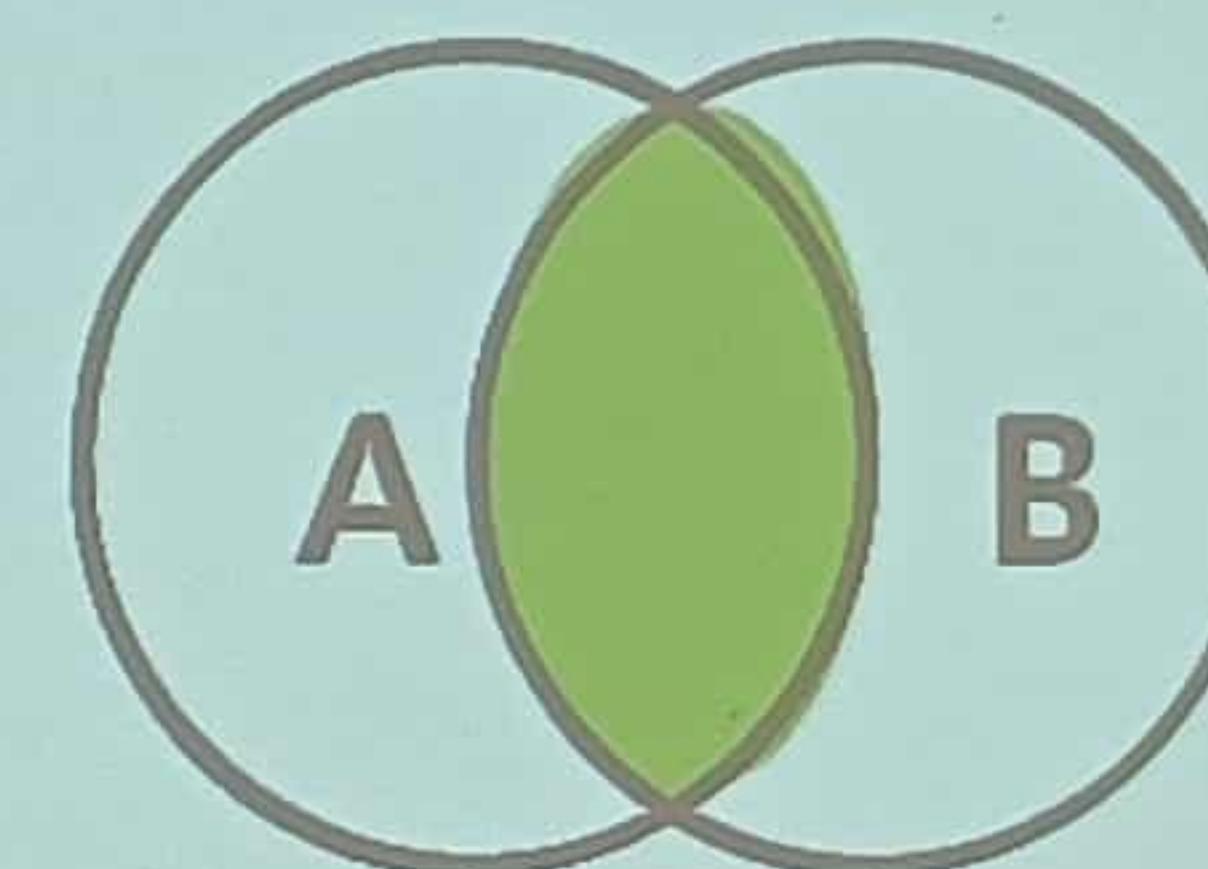


```
SELECT * FROM TableA A  
LEFT JOIN TableB B  
ON A.key = B.key;
```

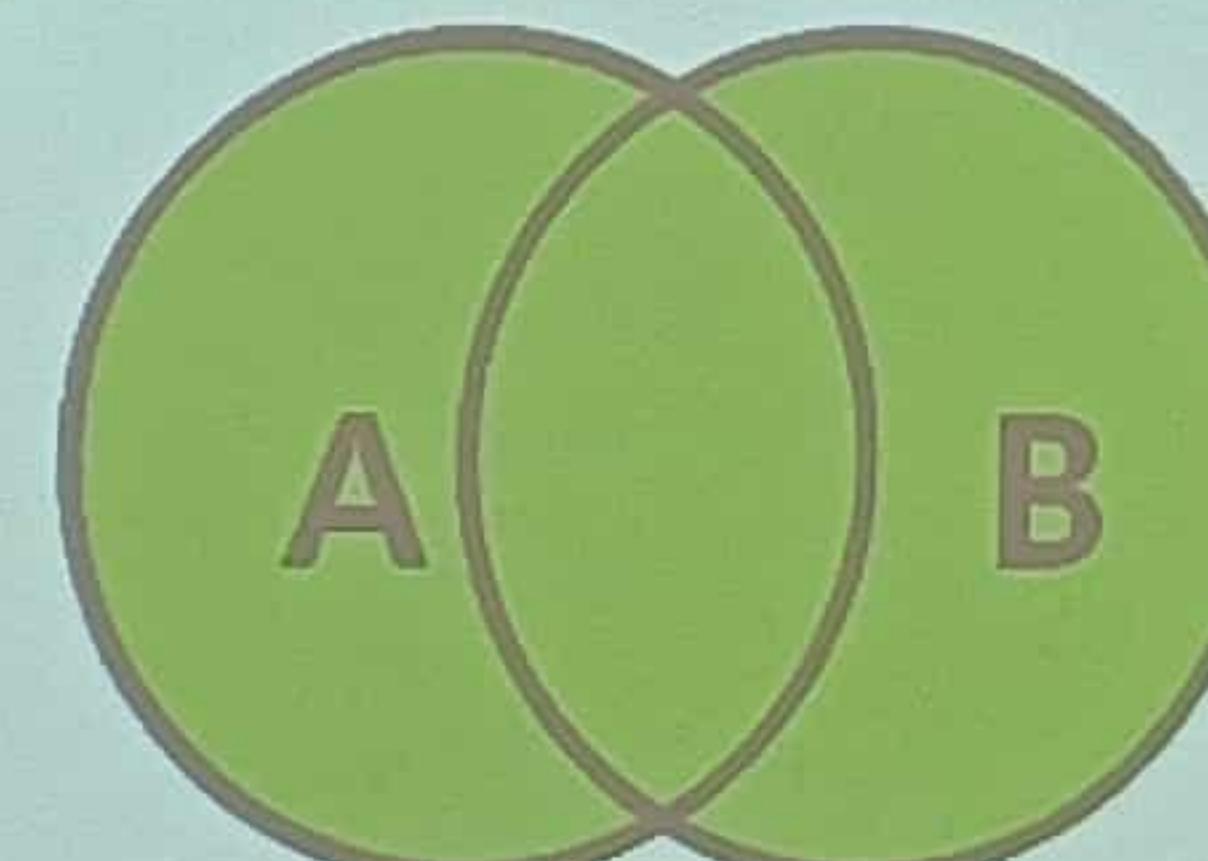


```
SELECT * FROM TableA A  
LEFT JOIN TableB B  
ON A.key = B.key  
WHERE B.Key IS NULL;
```

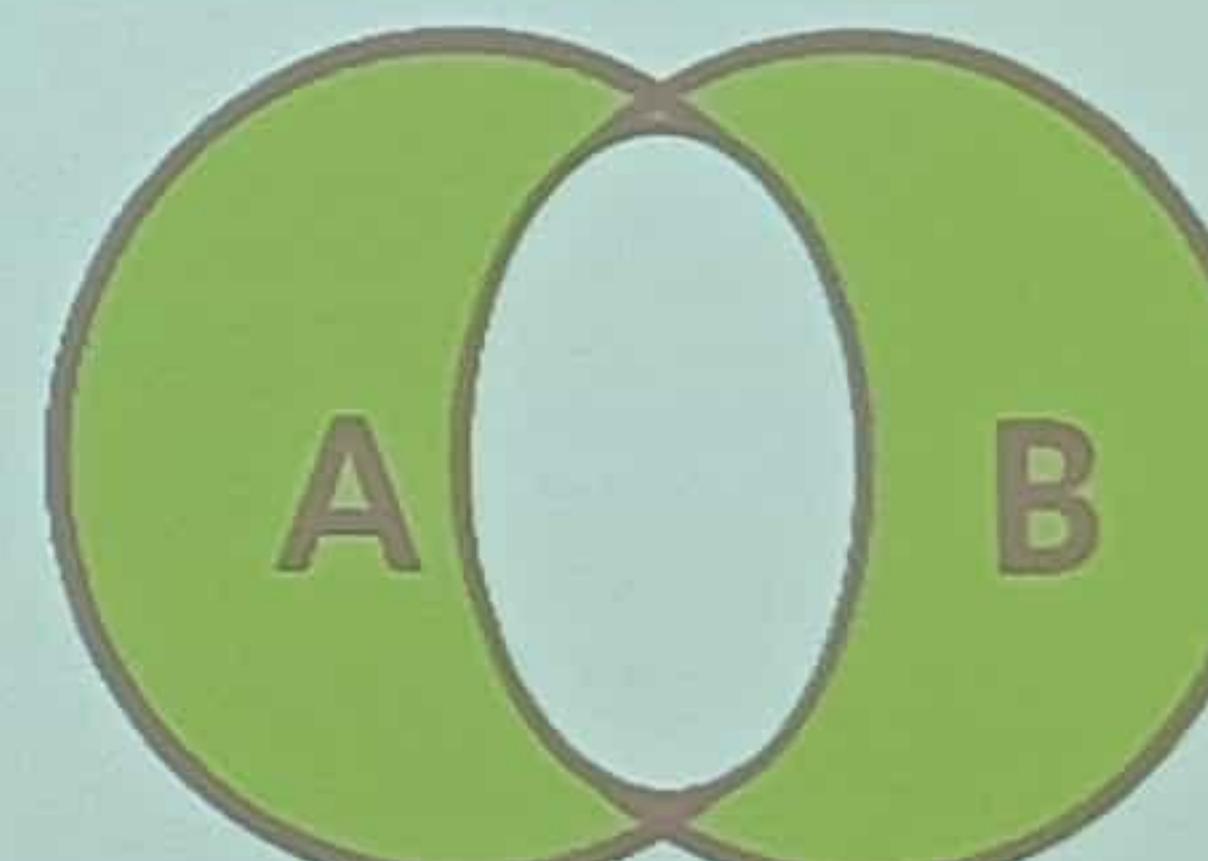
SQL JOIN



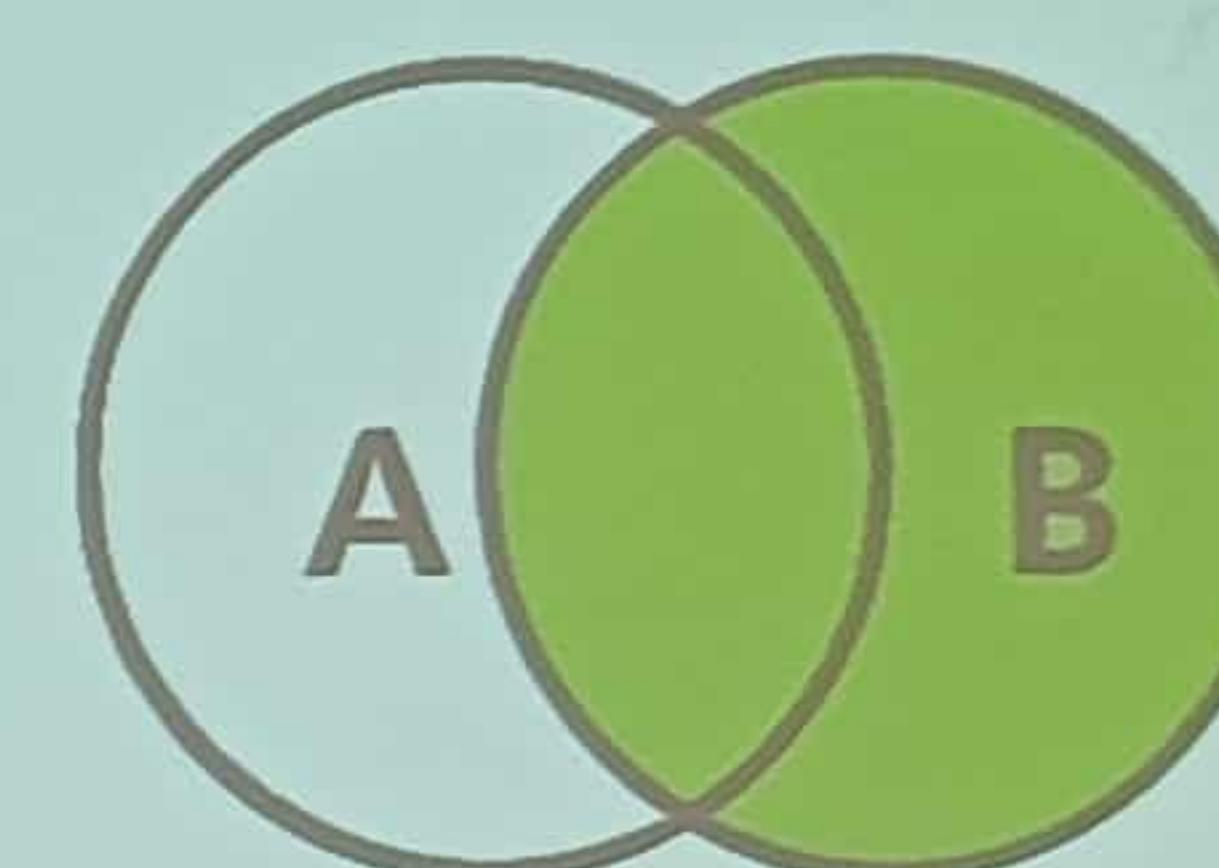
```
SELECT * FROM TableA A INNER JOIN TableB B ON A.key = B.key;
```



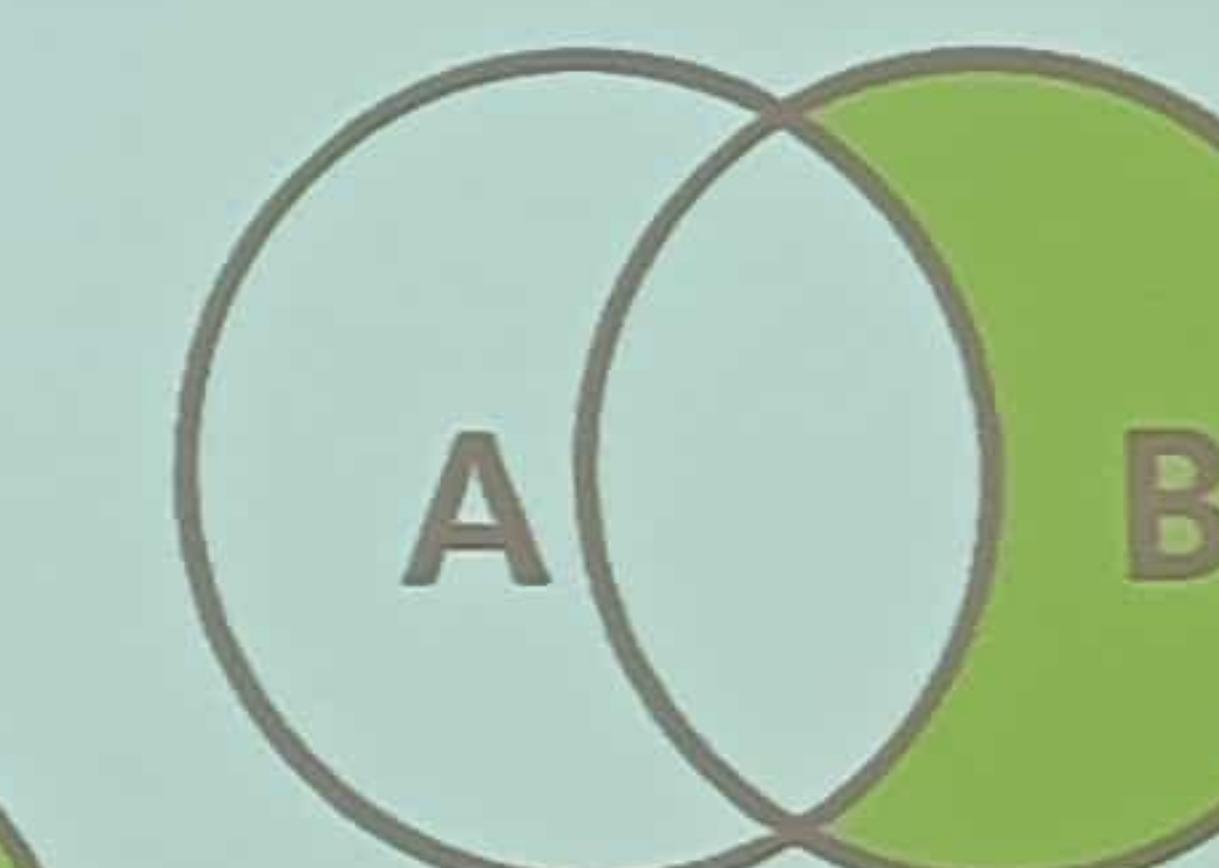
```
SELECT * FROM TableA A LEFT JOIN TableB B  
UNION  
SELECT * FROM TableA A RIGHT JOIN TableB B
```



```
SELECT * FROM TableA A LEFT JOIN TableB B  
UNION  
SELECT * FROM TableA A RIGHT JOIN TableB B  
WHERE A.key IS NULL OR B.key IS NULL
```



```
SELECT * FROM TableA A  
RIGHT JOIN TableB B  
ON A.key = B.key;
```



```
SELECT * FROM TableA A  
RIGHT JOIN TableB B  
ON A.key = B.key  
WHERE A.Key IS NULL;
```

number
id
iol
pre
Postges
SET

Where = 1 in E
any

SELECT 1;

px1