DB2 Assignments 2 + 3

DB2 Assignments 2

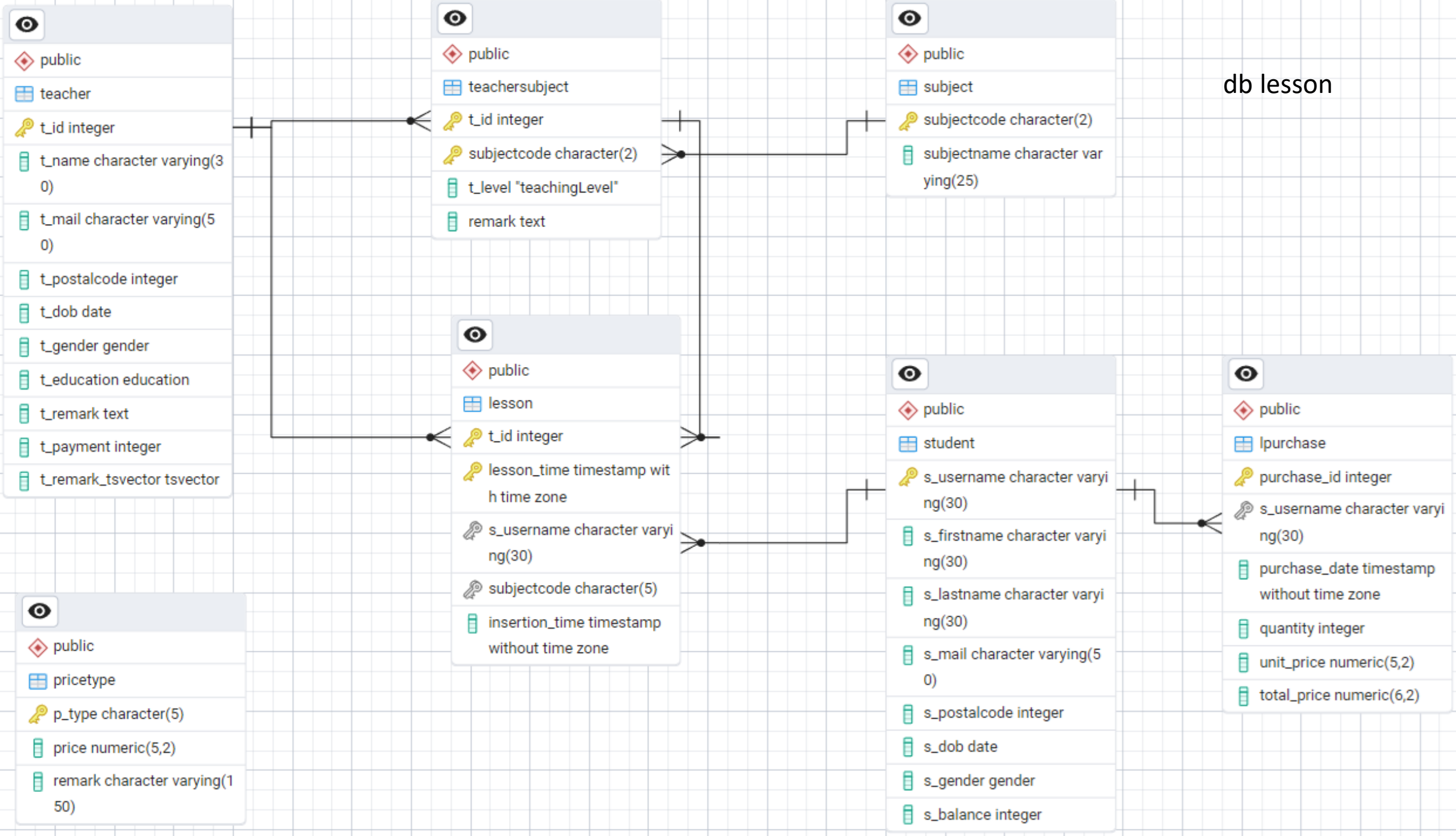
Lesson Database

Implement database lesson according to ERD on next slide.

ERD: ER-Diagram: visualized scheme of an existing database. An ERD is NOT a database design.

- add constraints (PKs, FKs) to fullfill integrity requirements,
- add check constraint to only allow purchases in increments of 5,
- insert values as provided in TEAMS,
- optional: add trigger to calculate price at purchase time depending on age of students,
    1. What event does the trigger get fired on?
    2. Is it a before or after trigger?
    3. Write a trigger function with the SQL code
    4. Create a trigger, associated with the specifc table and specific event
    5. Execute your trigger function within your trigger
    6. Test your trigger

eva.knirsch@kiu.edu.ge


public
teacher
t_id integer
t_name character varying(30)
t_mail character varying(50)
t_postalcode integer
t_dob date
t_gender gender
t_education education
t_remark text
t_payment integer
t_remark_tsvector tsvector
public
teachersubject
t_id integer
subjectcode character(2)
t_level "teachingLevel"
remark text
public
subject
subjectcode character(2)
subjectname character varying(25)
db lesson
public
lesson
t_id integer
lesson_time timestamp with time zone
s_username character varying(30)
subjectcode character(5)
insertion_time timestamp without time zone
public
student
s_username character varying(30)
s_firstname character varying(30)
s_lastname character varying(30)
s_mail character varying(50)
s_postalcode integer
s_dob date
s_gender gender
s_balance integer
public
lpurchase
purchase_id integer
s_username character varying(30)
purchase_date timestamp without time zone
quantity integer
unit_price numeric(5,2)
total_price numeric(6,2)
public
pricetype
p_type character(5)
price numeric(5,2)
remark character varying(150)

Referential Integrity

Insert the following row into table lesson:

```
INSERT INTO public.lesson(
            t_id, s_username, subjectcode)
            VALUES (1, 'Wolf', 'CH');
```

What error do you get and why?

eva.knirsch@kiu.edu.ge

Referential Integrity

We schedule a couple of lessons in our database.
What happens when we insert the following two rows into table lesson?

```
INSERT INTO lesson (t_id, lesson_time, s_username, subjectcode)
            VALUES (14, '2024-04-08 05:22:12.000000', 'Mickey', 'EN');

INSERT INTO lesson (t_id, lesson_time, s_username, subjectcode)
            VALUES (14, '2024-04-08 05:22:12.000000', 'Rose', 'EN');
```

Does the database react as we expect?

eva.knirsch@kiu.edu.ge

Assignments 3

PostGreSQL Transaction Example

Create the following table:

```
CREATE TABLE public.test
(
    "ID" serial,
    name character varying(30),
    PRIMARY KEY ("ID")
);

ALTER TABLE IF EXISTS public.test
    OWNER to postgres;
```


eva.knirsch@kiu.edu.ge

PostGreSQL Transaction Example

Run the following command sequence, in **PSQL**:

```
START TRANSACTION;
        INSERT into test values (1,'Miller');
        INSERT into test values (2,'Doe');
        ROLLBACK;
START TRANSACTION;
        INSERT into test values (3,'Neuer');
        INSERT into test  values (4,'Herrmann');
        COMMIT;
        INSERT into test values (5,'Niemann');
        INSERT into test  values (6,'Smith');
        rollback;
```

Which rows are in the table after running the sequence?
Explain!

eva.knirsch@kiu.edu.ge

PostGreSQL Transaction Example

- delete the data rows out of table test.
- run the same command sequence in **PGAdmin4**:

```
START TRANSACTION;
    INSERT into test values (1,'Miller');
    INSERT into test values (2,'Doe');
    ROLLBACK;
    START TRANSACTION;
    INSERT into test values (3,'Neuer');
    INSERT into test  values (4,'Herrmann');
    COMMIT;
    INSERT into test values (5,'Niemann');
    INSERT into test  values (6,'Smith');
    rollback;
```

https://www.postgresql.org/docs/current/tutorial-transactions.html

Some client libraries issue BEGIN and COMMIT commands automatically, so that you might get the effect of transaction blocks without asking. Check the documentation for the interface you are using.

Which rows are in the table after running the sequence?

Never use PGAdmin for transaction tests!

eva.knirsch@kiu.edu.ge

PostgreSQL Autocommit

Most RDBMS know two modes for transaction commits. The modes are controlled by the **autocommit** command.

- MariaDB: show variables like '%autocommit'
- PostgreSQL: \echo :AUTOCOMMIT

```
lesson=# \echo :AUTOCOMMIT
lesson=# \set AUTOCOMMIT off | on
```

In mariadb INNODB and PostgreSQL the default setting is: Autocommit = ON

By default (without BEGIN), PostgreSQL executes transactions in "autocommit" mode, that is, each statement is executed in its own transaction and a commit is implicitly performed at the end of the statement (if execution was successful, otherwise a rollback is done).
https://www.postgresql.org/docs/16/sql-begin.html

eva.knirsch@kiu.edu.ge

PostGreSQL Transaction Example

- Delete the data rows out of table test.
- Set autocommit OFF in command line
- Run the same command sequence again on command line

```
START TRANSACTION;
    INSERT into test values (1,'Miller');
    INSERT into test values (2,'Doe');
    ROLLBACK;
    START TRANSACTION;
    INSERT into test values (3,'Neuer');
    INSERT into test  values (4,'Herrmann');
    COMMIT;
    INSERT into test values (5,'Niemann');
    INSERT into test  values (6,'Smith');
    rollback;
```

Which rows are in the table after running the same sequence in PSQL but with autocommit off?

eva.knirsch@kiu.edu.ge

PostGreSQL Transaction Example

- Delete the data rows out of table test.
- Run the same command sequence again on command line –
  but without the last rollback command!

```
START TRANSACTION;
        INSERT into test values (1,'Miller');
        INSERT into test values (2,'Doe');
        ROLLBACK;
        START TRANSACTION;
        INSERT into test values (3,'Neuer');
        INSERT into test  values (4,'Herrmann');
        COMMIT;
        INSERT into test values (5,'Niemann');
        INSERT into test  values (6,'Smith');
```

- Check that transaction is still active:
  SELECT * FROM pg_stat_activity WHERE state IN ('active', 'idle in transaction');
- end transaction manually by typing command commit; in PSQL
- Check, which rows are in the table

eva.knirsch@kiu.edu.ge

PostgreSQL Autocommit

On the server side, PostgreSQL operates in autocommit mode.

On the client side, autocommit mode can be turned off in two ways:

1. Explicitly: By setting autocommit OFF
2. Implicitly: By starting a transaction with a BEGIN statement

If you disable autocommit, Postgres understands all commands as implicit transactions – independent of whether you explicitly start a transaction or not.

But: PostgreSQL will wait for a commit / rollback to finish a transaction.
If that command does not come, the session will remain in the state “idle in transaction”. This means that locks may not be released.

eva.knirsch@kiu.edu.ge

Transactions on Different Rows

- Open 2 sessions in PSQL (command line sessions)
- Run two concurrent transactions that operate on the same table (teacher) but **different rows.**
- Run the transactions in default isolation level

T1: reads out t_payment counter of **teacher 1**, then sleeps for some seconds, then sets payment counter to 0

T2: starts right after T1, increments t_payment counter of **teacher 2** by 3, then commits

- Do both transactions run right through and commit?
- Change isolation level to repeatable read.
- Do both transactions run through and commit?
- What conclusions do you draw regarding PostgreSQL isolation mechanism?

eva.knirsch@kiu.edu.ge

Transactions on Different Colums

- Open 2 sessions in PSQL (command line sessions)
- Run two concurrent transactions that operate on the same table (teacher) on the same table (teacher) and on **the same row** but on **different columns**.
- Run the transactions in **default isolation level**

T1: reads out t_payment counter of teacher 2 , then does some application logic, then resets the counter of teacher 2 to 0

T2 updates t_education of teacher 2, sets t_education = 'Bachelor'

- Do both transactions run right through and commit?
- Can both transactions eventually commit?

eva.knirsch@kiu.edu.ge

Transactions on Different Columns

- Change the isolation level to repeatable read:

T1: reads out t_payment counter of teacher 2 , then does some application logic, then resets the counter of teacher 2 to 0

T2 updates t_education of teacher 2, sets t_education = 'Bachelor'

- Do both transactions run right through and commit?
- Can both transactions eventually commit?

- Let T2 rollback instead of commit. What effect does this have on T1?

eva.knirsch@kiu.edu.ge

Concurrent Write Transactions

Run the 2 sequences of sliude 21 again, in default isolation level. The only difference is that T2 now also does some application code – simulated by select pg_sleep() – **see red line**

```
/* client session 1 – Agency pays teacher 1*/
Begin;
select t_payment from teacher where t_id = 1;
/* some application code - calculating how much money teacher gets and doing the money transfer */
select pg_sleep(20);
update teacher set t_payment=0 where t_id = 1;
commit;

/* Client Session 2 – teacher reports back hours  */
Begin;
update teacher set t_payment = t_payment + 3 where t_id = 1;
SELECT t_payment FROM teacher WHERE t_id = 1;
select pg_sleep(20);
commit;
```

Explain the execution.

eva.knirsch@kiu.edu.ge

Isolation Levels – feel free to use AI for your answer.

1. Which different isolation levels does the SQL standard have?

2. What anomalies (consistency problems) do the different levels prevent?

3. Which isolation levels does Postgres offer?

4. Which isolation levels does MySQL / MariaDB offer?

eva.knirsch@kiu.edu.ge