1222·2022 800 ANNI

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Foundations of Databases A.Y. 2021-2022

Master Degree in
ICT for Internet and Multimedia

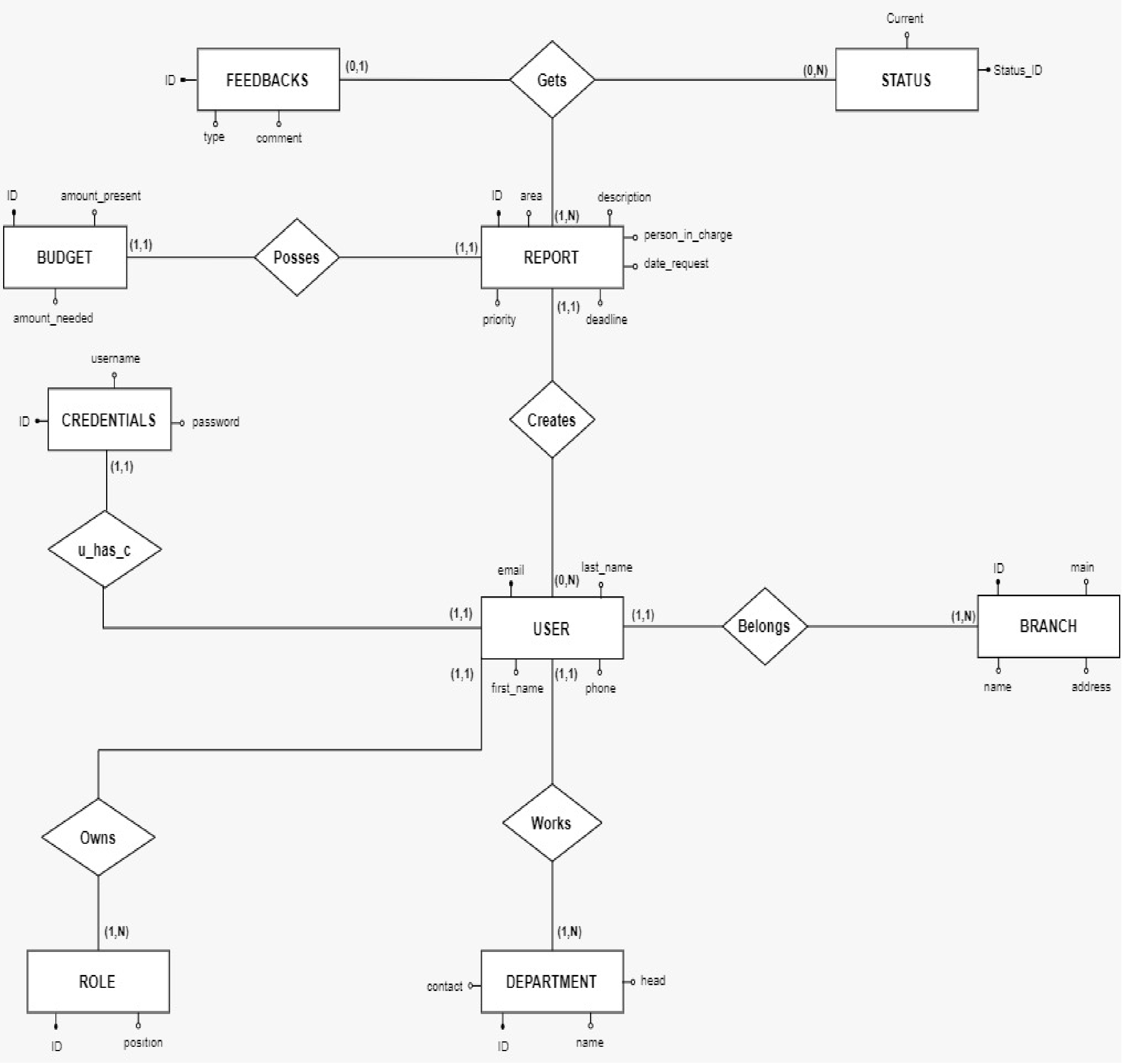
Homework 3 – Physical Design

Group Name: RD
Project: non profit organizations

Last Name	***First Name***	***Student Num***
ATES	*HAKAN*	*2049550*
CICEK	*KADER*	*2049526*
LOKO	*ISIDORUS MAU*	*2040498*
MURDAYAN	*YEKATERINA*	*2049529*
OZTURK	*MEHMET*	*2049527*
PAJAJ	*XHEINA*	*2055575*
PRAVISHI	*GAURI*	*2041448*
SHOFI	*ALFI BAQIATUS*	*2041377*
ZARE	*MOHAMMAD MEHDI*	*2041379*
ZIU	*VANESA*	*2050797*

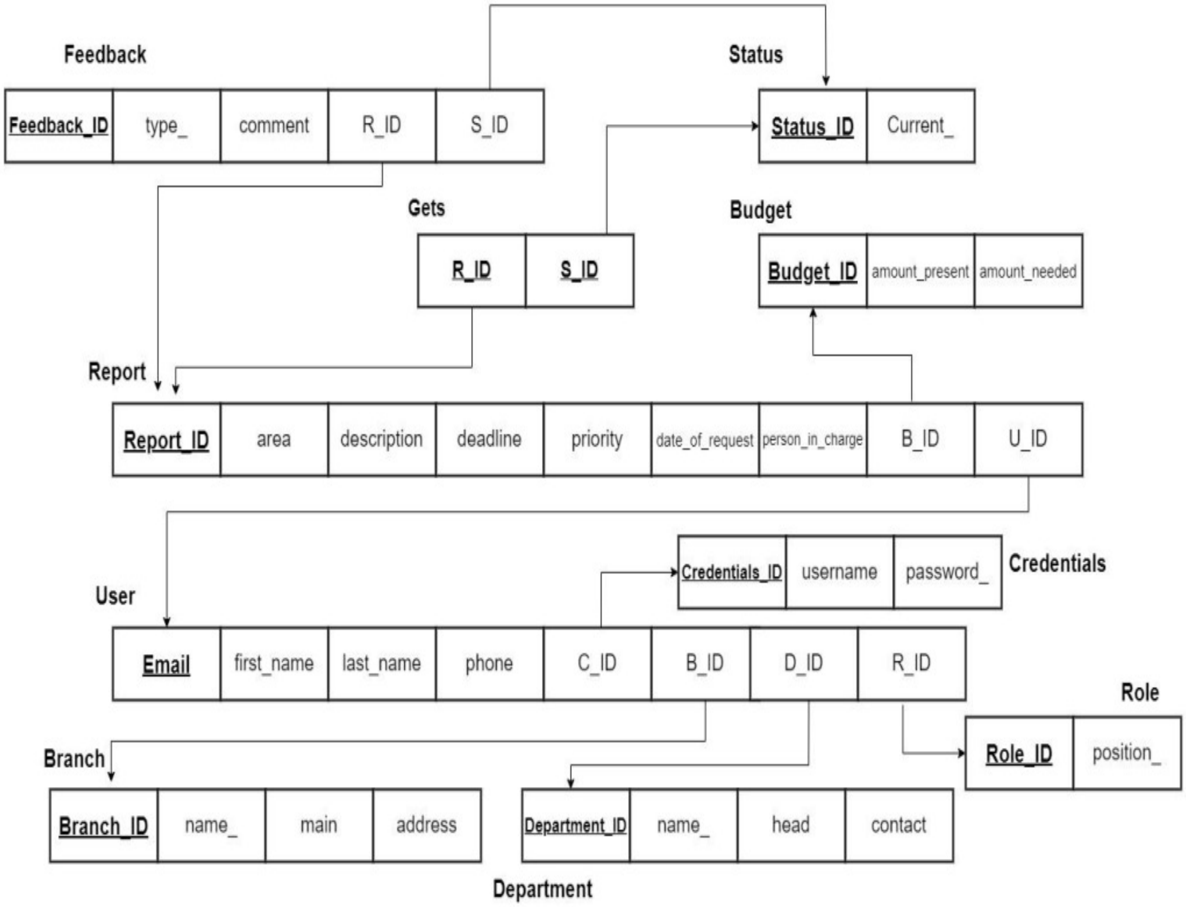
Changes on ER Schema

We have changed the ER Schema and Relational Schema based on the comment professor added on Bitbucket. Respectively, we have changed the cardinality of Creates relationship on User side and cardinality of relationship Gets on Status side. We have created a triality relationship between Feedback, Status and Report entities. The Permission entity is already removed too. On Relational Schema everything is mapped from the changes done in ER Schema.



Variations to the Relational Schema

Figure 1 shows the relational schema. To reduce the workload on the system, we created a separate place called credentials to use in the login section.W e wanted to leave the event as it is because we did not want it to interfere with the user . Feedback has also been connected to report and status We removed the date in the status and Feedback.. The Permission has been removed. The Gets attribute added for getting status and feedback to the report .And a lot of variables have been renamed so that it can be understood more easily .



Physical Schema

```
CREATE DATABASE DB1920;
CREATE SCHEMA D1;

CREATE DOMAIN D1.Comment AS VarChar(200)
    CONSTRAINT propercomment CHECK (((VALUE):: text ~* ’(\w+)’:: text));

CREATE DOMAIN D1.Description AS VarChar(1000);
CREATE DOMAIN D1.Email AS VarChar(2000)
CONSTRAINT properemail CHECK (((VALUE)::
text ˜* ’ˆ(.+)@(.+)$’:: text));

CREATE DOMAIN D1.Username AS VarChar(1000)
    CONSTRAINT properusername CHECK (((VALUE)::
                                          text ˜* ’[a-zA-Z]{7,11}’:: text));

CREATE DOMAIN D1.Password_ AS VarChar(1000)
    CONSTRAINT properpassword CHECK (((VALUE)::
                                          text ˜* ’(\S{11,18})’:: text));

CREATE DOMAIN D1.Phone AS VarChar(100)
    CONSTRAINT properphone CHECK (((VALUE) ::
                                       text ˜* ’(\d+)’ :: text));

CREATE TYPE D1.Position_
AS ENUM (’Branch Office Team’,
          ’Financial Team’,
          ’Coordinators team’,
          ’Generalate Main Office’ );

CREATE TYPE D1.Type_
AS ENUM (’Positive’,
    ’Negative’,
    ’Neutral’);

CREATE TYPE D1.D_Name
AS ENUM (’Finance’,
          ’Logistic’,
          ’HR’,
          ’Customer Care’,
  ’IT’);

CREATE TABLE D1.Status(Status_ID UUID PRIMARY KEY ,
                       Current_ Boolean NOT NULL );

CREATE TABLE D1.Budget(Budget_ID UUID PRIMARY KEY ,
                       Amount_Present MONEY, Amount_Needed MONEY NOT NULL);

CREATE TABLE D1.Credentials(Credentials_ID UUID PRIMARY KEY ,
                            Username D1.Username NOT NULL ,
                            Password_ D1.Password_ NOT NULL);

CREATE TABLE D1.Role(Role_ID UUID PRIMARY KEY ,
                     Position_ D1.Position_ NOT NULL);

CREATE TABLE D1.Department(Department_ID UUID PRIMARY KEY ,
                           Name_ Varchar(200) NOT NULL,
                           Head_ Varchar(200) NOT NULL,
                           Contact D1.Phone );

CREATE TABLE D1.Branch(Branch_ID UUID PRIMARY KEY ,
                       Name_ Varchar(200) NOT NULL,
                       Main_ BOOLEAN NOT NULL, Address VarChar(1000) NOT NULL);

CREATE TABLE D1.User(Email D1.Email PRIMARY KEY ,
                     First_Name Varchar(200) NOT NULL,Last_Name Varchar(200) NOT NULL,
                     Phone D1.Phone NOT NULL, C_ID UUID NOT NULL,B_ID UUID NOT NULL,R_ID UUID NOT NULL,
                     D_ID UUID NOT NULL,FOREIGN KEY (C_ID) REFERENCES D1.Credentials(Credentials_ID),
                     FOREIGN KEY (B_ID) REFERENCES D1.Branch(Branch_ID),
                     FOREIGN KEY (R_ID) REFERENCES D1.Role(Role_ID),
                     FOREIGN KEY (D_ID) REFERENCES D1.Department(Department_ID) );

CREATE TABLE D1.Report(Report_ID UUID PRIMARY KEY ,
                       Area Varchar(200) NOT NULL,Description D1.Description NOT NULL,
                       Deadline Date NOT NULL,Person_in_charge Varchar(200) NOT NULL,
                       B_ID UUID NOT NULL,U_ID D1.Email NOT NULL,
                       FOREIGN KEY (B_ID) REFERENCES D1.Budget(Budget_ID),
                       FOREIGN KEY (U_ID) REFERENCES D1.User(Email) );


ALTER TABLE D1.Report ADD Date_of_request DATE nOT null Default CURRENT_DATE;
ALTER TABLE D1.Report ADD Priority INT generated always as  ((DATE_PART(’year’, Deadline) - DATE_PART(’y
* 12 + (DATE_PART(’month’, Deadline) - DATE_PART(’month’, Date_of_request)))  STORED;

CREATE TABLE D1.Feedback(Feedback_ID UUID PRIMARY KEY ,
                         Type_ D1.Type_ NOT NULL, Comment D1.Comment NOT NULL,
                         R_ID UUID NOT NULL,
                         S_ID UUID NOT NULL,FOREIGN KEY (R_ID) REFERENCES D1.Report(Report_ID),
                         FOREIGN KEY (S_ID) REFERENCES D1.Status(Status_ID) );

CREATE TABLE D1.Gets( R_ID UUID NOT NULL,S_ID UUID NOT NULL,
                      FOREIGN KEY (R_ID) REFERENCES D1.Report(Report_ID),
                      FOREIGN KEY (S_ID) REFERENCES D1.Status(Status_ID));
```

Populate the Database: Example

```
INSERT INTO D1.Status(Status_ID,Current_) VALUES
(’e3abfd3e-2c03-4288-84bf-87a817c4de9b’, ’True’),
(’b6ec1f04-c03d-4b24-9f5c-7917d18629fa’, ’True’),
(’f5739f25-45d6-495d-82b1-2ca437ec7415’, ’True’),
(’224049cf-4a70-4a2a-9151-a677f07258fc’, ’True’),
(’f74e5ef8-16ff-44ad-aa05-0fdc1a6a8c12’, ’True’),
(’bd49a692-d1ee-45a2-bd32-6e29e1aa884d’, ’True’),
(’0b1d6916-f732-400a-82a5-9e5c47448435’, ’True’),
(’ff9f614b-1b67-4c7f-8f14-d1b244b98aa0’, ’True’);

INSERT INTO D1.Budget(Budget_ID,Amount_Present,Amount_Needed) VALUES
(’efe47302-d960-4749-9260-e0857799b04d’, ’25000’, ’24000’),
(’9cd38ead-cbdc-4473-9169-bf315a8d46bc’, ’75000’, ’64000’),
(’1f119843-61e7-4359-9266-17bf809ad802’, ’3700’, ’39000’),
(’7727661b-04d2-44be-b65c-a64bbb453934’, ’83000’, ’63000’),
(’bdd7b3f4-e1a0-4983-b3e8-bf32719a425f’, ’1700’, ’32000’),
(’e9830d6e-c38c-46f5-980e-246c2044f8e9’, ’0’, ’39000’);

INSERT INTO D1.Budget(Budget_ID,Amount_Needed) VALUES
(’af83b685-c9de-4bf4-8910-ae0564ed032b’, ’100000’);
```

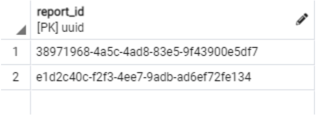
```
INSERT INTO D1.Credentials(Credentials_ID,Username,Password_) VALUES
(’03afe1c4-d723-4a9a-ba5f-d970d7ab85b7’, ’lenasmith’, ’marco*125_marco’),
(’f1703d8c-2241-44e2-9de2-3b92b48b4c35’, ’nicolgreen’, ’512512124355’),
(’da7725aa-233b-43e9-87cc-9401de8ec873’, ’tomtaylor’, ’387191*ggwp’),
(’9f42561e-02e9-4593-9a42-cd9ed4b7ffaa’, ’paulmiller’, ’imnotaspy_123*’),
(’56ebebd5-8c7a-45f4-8b7a-38ee4f650967’, ’alenlee’, ’387191*ggwp’),
(’a166313c-2253-4f4f-9b9e-aba819f2feea’, ’davidking’, ’32451kingda’),
(’73274d5a-bd0f-456c-a4fd-3b3962731779’, ’obmarwood’, ’191*obmarrrr’),
(’6a6a2a07-ea3e-4299-9ef2-6b33bd7a163e’, ’jonesmith’, ’jones4444432’),
(’ca9929d1-0c96-486a-980b-2dcb4bf26b36’, ’lucywhite’, ’3treesontheway’),
(’dbf1d89d-5bb1-4b7c-b95f-a321160ae792’, ’lonakley’, ’dontthinkaboutit’);


INSERT INTO D1.Role(Role_ID,Position_) VALUES
(’f1e0f5d2-8b5f-4c73-b473-41fd68acc94d’, ’Generalate Main Office’),
(’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’, ’Branch Office Team’),
(’3ddd515c-0232-44a9-9e3a-9c342e13f948’, ’Financial Team’),
(’53b45115-11d1-476b-98a6-b5e11fe2a079’, ’Coordinators team’);


INSERT INTO D1.Department(Department_ID,Name_,Head_ ,Contact) VALUES
(’2df07ce3-8042-4821-862f-a90e29b57da0’, ’Customer Care’, ’Marco Tomasin’, ’00391234567890’),
(’74ad40b1-56ec-4d08-81cc-baa151963a07’, ’HR’, ’Marry Tommy’, ’00391234567891’),
(’de9cc2b4-e422-4bba-9832-d59b7310058b’, ’Finance’, ’Katty Jones’, ’00391234567893’),
(’563f4fc5-3b87-46ac-ad85-e9aceb04c549’, ’Logistic’, ’Martin Smith’, ’00391234567897’),
(’b482c32e-8b72-4d73-be34-3ae17bc0870e’, ’IT’, ’Adele Dan’, ’00391234567897’);


INSERT INTO D1.Branch(Branch_ID,Name_,Main_,Address) VALUES
(’79cef980-631e-4288-b223-801a9bd04128’, ’Rome Office’, ’True’, ’Via Pietro Liberi 1, Rome’),
(’dba7366d-7b0a-47ef-82e6-e7b0cc2c8d6e’, ’Padova Office’, ’False’, ’Piazza Duomo 11, Padova’),
(’53db79cf-e902-411c-ac52-9a14ce649baf’, ’TelAviv Office’, ’False’, ’Street Telaviv 1, Tel Aviv’),
(’3b544b74-d728-4137-8b32-eb44f9581c59’, ’Tirana Office’, ’False’,  ’Street Zogu I, Prane Redaksise 1’),
(’2389977b-e44b-41d2-835c-6e2dce3b7c4e’, ’Moskow Office’, ’False’, ’Ul Voloshinoy, bld 6/ appt 51’),
(’262aa3a6-2fe2-4a22-aa5b-4a4797d3d874’, ’Berlin Office’, ’False’, ’Unter den Linden 15’),
(’9d510dab-34b4-4dd9-a64d-a5e716355acf’, ’London Office’, ’False’, ’33 Nottingham Rd’);
;

INSERT INTO D1.User(Email,First_Name,Last_Name,Phone,C_ID,B_ID,D_ID,R_ID) VALUES
(’lena.smith84@gmail.com’, ’Lena’, ’Smith’,  ’00391234567890’,
’03afe1c4-d723-4a9a-ba5f-d970d7ab85b7’, ’79cef980-631e-4288-b223-801a9bd04128’,
’2df07ce3-8042-4821-862f-a90e29b57da0’,  ’f1e0f5d2-8b5f-4c73-b473-41fd68acc94d’),
(’nicol.green77@gmail.com’, ’Nicol’, ’Green’,  ’00391234567891’,
’f1703d8c-2241-44e2-9de2-3b92b48b4c35’, ’79cef980-631e-4288-b223-801a9bd04128’,
’74ad40b1-56ec-4d08-81cc-baa151963a07’, ’f1e0f5d2-8b5f-4c73-b473-41fd68acc94d’),
(’tomtaylor85@gmail.com’, ’Tom’, ’Taylor’,  ’00391234567893’,
’da7725aa-233b-43e9-87cc-9401de8ec873’, ’79cef980-631e-4288-b223-801a9bd04128’,
’de9cc2b4-e422-4bba-9832-d59b7310058b’, ’3ddd515c-0232-44a9-9e3a-9c342e13f948’),
(’paulmiller@gmail.com’, ’Paul’, ’Miller’,  ’00391234567897’,
’9f42561e-02e9-4593-9a42-cd9ed4b7ffaa’, ’79cef980-631e-4288-b223-801a9bd04128’,
’563f4fc5-3b87-46ac-ad85-e9aceb04c549’, ’53b45115-11d1-476b-98a6-b5e11fe2a079’),
(’alenlee@gmail.com’, ’Alen’, ’Lee’,  ’00394443325567’,
’56ebebd5-8c7a-45f4-8b7a-38ee4f650967’, ’79cef980-631e-4288-b223-801a9bd04128’,
’b482c32e-8b72-4d73-be34-3ae17bc0870e’,  ’53b45115-11d1-476b-98a6-b5e11fe2a079’),
(’davidking77@gmail.com’, ’David’, ’King’,  ’00393656547888’,
’a166313c-2253-4f4f-9b9e-aba819f2feea’, ’dba7366d-7b0a-47ef-82e6-e7b0cc2c8d6e’,
’563f4fc5-3b87-46ac-ad85-e9aceb04c549’, ’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’),
(’obmarwood@gmail.com’, ’Obmar’, ’Wood’,  ’00396575674355445’,
’73274d5a-bd0f-456c-a4fd-3b3962731779’, ’53db79cf-e902-411c-ac52-9a14ce649baf’,
’2df07ce3-8042-4821-862f-a90e29b57da0’, ’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’),
(’jonesmith@gmail.com’, ’Jone’, ’Smith’,  ’0039747543243667878’,
’6a6a2a07-ea3e-4299-9ef2-6b33bd7a163e’, ’3b544b74-d728-4137-8b32-eb44f9581c59’,
’2df07ce3-8042-4821-862f-a90e29b57da0’, ’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’),
(’lucy.white85@gmail.com’, ’Lucy’, ’White’,  ’003912543893’,
’ca9929d1-0c96-486a-980b-2dcb4bf26b36’, ’2389977b-e44b-41d2-835c-6e2dce3b7c4e’,
’de9cc2b4-e422-4bba-9832-d59b7310058b’, ’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’),
(’lonakley93@gmail.com’, ’Lona’, ’Kley’,  ’00391234564397’,
’dbf1d89d-5bb1-4b7c-b95f-a321160ae792’, ’262aa3a6-2fe2-4a22-aa5b-4a4797d3d874’,
’563f4fc5-3b87-46ac-ad85-e9aceb04c549’, ’7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e’);

INSERT INTO D1.Report(Report_ID,Area,Description,Deadline,Person_in_charge,B_ID,U_ID) VALUES
(’2714f5d8-d906-4c80-8898-d85d777cf2e2’, ’Europe’, ’For people who need for health’,
’2022/12/12’, ’Marco Tomasin’, ’efe47302-d960-4749-9260-e0857799b04d’, ’davidking77@gmail.com’),
(’38971968-4a5c-4ad8-83e5-9f43900e5df7’, ’Africa’, ’For Church of one of Asian’,’2022/03/02’,
’Marry Tommy’, ’9cd38ead-cbdc-4473-9169-bf315a8d46bc’,’obmarwood@gmail.com’),
(’e1d2c40c-f2f3-4ee7-9adb-ad6ef72fe134’, ’Asia’, ’For school books’,’2022/07/12’,
’Katy George’, ’1f119843-61e7-4359-9266-17bf809ad802’, ’obmarwood@gmail.com’),
(’fab087e8-8cd7-482e-96e4-bb2cebc54ecf’, ’Asia’, ’For people cant pay their surgery’,
’2022/10/20’, ’Rihanna Jones’, ’7727661b-04d2-44be-b65c-a64bbb453934’, ’lucy.white85@gmail.com’),
(’46d9f71e-18f7-42c3-8199-80c9d23ce1ed’, ’Europe’, ’For people who need for health’,
’2022/11/01’,  ’Marco Tomasin’, ’bdd7b3f4-e1a0-4983-b3e8-bf32719a425f’, ’jonesmith@gmail.com’),
(’0c603099-c801-47c6-9abb-604e9665b912’, ’Africa’, ’For Church of one of Asian’,
’2022/09/02’, ’Marry Tommy’, ’e9830d6e-c38c-46f5-980e-246c2044f8e9’,’lonakley93@gmail.com’),
(’2cd38322-b108-42f1-b410-7bd25d842cdb’, ’Asia’, ’For school books’,’2022/09/12’,
’Katy George’, ’af83b685-c9de-4bf4-8910-ae0564ed032b’, ’davidking77@gmail.com’);



INSERT INTO D1.Feedback(Feedback_ID,Type_,Comment,R_ID,S_ID) VALUES
(’e87fcb16-af1b-45a4-9d21-651279b53d97’, ’Negative’, ’Need to finish soon’,
’2714f5d8-d906-4c80-8898-d85d777cf2e2’, ’e3abfd3e-2c03-4288-84bf-87a817c4de9b’),
(’b10f0326-cfcd-4b26-9ba5-a99e06c4b20f’, ’Neutral’, ’Good work’,
’38971968-4a5c-4ad8-83e5-9f43900e5df7’,  ’b6ec1f04-c03d-4b24-9f5c-7917d18629fa’),
(’ec5bdf63-6d65-48ad-9256-a61483d0f4a6’, ’Positive’, ’Need more amount’,
’e1d2c40c-f2f3-4ee7-9adb-ad6ef72fe134’,’f5739f25-45d6-495d-82b1-2ca437ec7415’),
(’89ab26d1-7032-4358-805f-06dd484a0e04’, ’Positive’, ’Need to be clear about country’,
’fab087e8-8cd7-482e-96e4-bb2cebc54ecf’, ’224049cf-4a70-4a2a-9151-a677f07258fc’);


INSERT Into D1.Gets(R_ID,S_ID) VALUES
(’2714f5d8-d906-4c80-8898-d85d777cf2e2’, ’e3abfd3e-2c03-4288-84bf-87a817c4de9b’),
(’38971968-4a5c-4ad8-83e5-9f43900e5df7’, ’b6ec1f04-c03d-4b24-9f5c-7917d18629fa’),
(’e1d2c40c-f2f3-4ee7-9adb-ad6ef72fe134’, ’f5739f25-45d6-495d-82b1-2ca437ec7415’),
(’fab087e8-8cd7-482e-96e4-bb2cebc54ecf’, ’224049cf-4a70-4a2a-9151-a677f07258fc’),
(’46d9f71e-18f7-42c3-8199-80c9d23ce1ed’, ’f74e5ef8-16ff-44ad-aa05-0fdc1a6a8c12’),
(’0c603099-c801-47c6-9abb-604e9665b912’, ’bd49a692-d1ee-45a2-bd32-6e29e1aa884d’),
('2cd38322-b108-42f1-b410-7bd25d842cdb', '0b1d6916-f732-400a-82a5-9e5c47448435'),
('46d9f71e-18f7-42c3-8199-80c9d23ce1ed', 'ff9f614b-1b67-4c7f-8f14-d1b244b98aa0');
```

Principal Queries

In this section we report the queries needed to perform some operations on the database:

1. (a) Print the list of Reports that have the priority equal or less than 8 and the deadline before 2023.03.10.

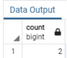
```
SELECT (Report.Report_ID)
FROM D1.Report
WHERE Report.Priority <= 8 AND Report.Deadline < '2023.03.10';
```



	report_id [PK] uuid
1	38971968-4a5c-4ad8-83e5-9f43900e5df7
2	e1d2c40c-f2f3-4ee7-9adb-ad6ef72fe134

(b) Find the number of Reports that have the priority equal or less than 8 and the deadline before 2023.03.10.

```
SELECT Count(Report.Report_ID)
FROM D1.Report
WHERE Report.Priority <= 8 AND Report.Deadline < '2023.03.10';
```

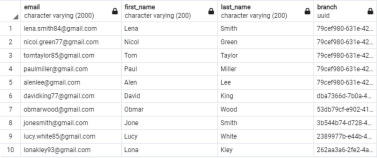


Data Output

	count bigint
1	2

2. Retrieve a list of users together with their related information email, first_name, last_name, togther with the Branch_ID they belong to using INNER JOIN.

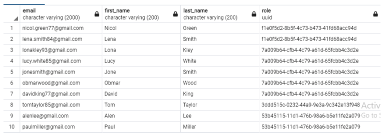
```
SELECT u.email, u.first_name, u.last_name, b.Branch_ID as Branch
FROM D1.User as u
INNER JOIN D1.Branch as b ON u.B_ID= b.Branch_ID;
```



	email character varying (2000)	first_name character varying (200)	last_name character varying (200)	branch uuid
1	lena.smith84@gmail.com	Lena	Smith	79cef980-631e-42...
2	nicol.green77@gmail.com	Nicol	Green	79cef980-631e-42...
3	tomtaylor85@gmail.com	Tom	Taylor	79cef980-631e-42...
4	paulmiller@gmail.com	Paul	Miller	79cef980-631e-42...
5	alenlee@gmail.com	Alen	Lee	79cef980-631e-42...
6	davidking77@gmail.com	David	King	dba7366d-7b0a-4...
7	obmarwood@gmail.com	Obmar	Wood	53db79cf-e902-41...
8	jonesmith@gmail.com	Jone	Smith	3b544b74-d728-4...
9	lucy.white85@gmail.com	Lucy	White	2389977b-e44b-4...
10	lonakley93@gmail.com	Lona	Kley	262aa3a6-2fe2-4a...

3. Retrieve a list of users together with their related information email, first_name, last_name, together with their corresponding ROLE using LEFT JOIN.

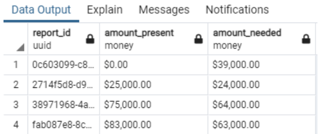
```
SELECT u.Email, u.first_name, u.last_name, r.Role_ID as Role
FROM D1.User as u
LEFT JOIN D1.Role as r ON u.R_ID = r.Role_ID
```



	email character varying (2000)	first_name character varying (200)	last_name character varying (200)	role uuid
1	nicol.green77@gmail.com	Nicol	Green	f1e0f5d2-8b5f-4c73-b473-41fd68acc94d
2	lena.smith84@gmail.com	Lena	Smith	f1e0f5d2-8b5f-4c73-b473-41fd68acc94d
3	lonakley93@gmail.com	Lona	Kley	7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e
4	lucy.white85@gmail.com	Lucy	White	7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e
5	jonesmith@gmail.com	Jone	Smith	7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e
6	obmarwood@gmail.com	Obmar	Wood	7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e
7	davidking77@gmail.com	David	King	7a009b64-cfb4-4c79-a61d-65fcbb4c3d2e
8	tomtaylor85@gmail.com	Tom	Taylor	3ddd515c-0232-44a9-9e3a-9c342e13f948
9	alenlee@gmail.com	Alen	Lee	53b45115-11d1-476b-98a6-b5e11fe2a079
10	paulmiller@gmail.com	Paul	Miller	53b45115-11d1-476b-98a6-b5e11fe2a079

4. Retrieve a list of reports together with their report id, present budget amount needed which their present budget is more than amount needed. Also retrieve those which their present amount is below 1000.

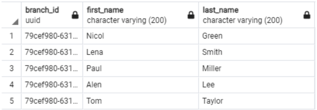
```
SELECT report_id, amount_present, amount_needed
FROM D1.report AS R INNER JOIN D1.budget as B
ON R.b_id = B.budget_id
WHERE amount_present > amount_needed
UNION
SELECT report_id, amount_present, amount_needed
FROM D1.report AS R INNER JOIN D1.budget as B
ON R.b_id = B.budget_id
WHERE amount_present::numeric::int < 1000
```



	report_id uuid	amount_present money	amount_needed money
1	0c603099-c8...	$0.00	$39,000.00
2	2714f5d8-d9...	$25,000.00	$24,000.00
3	38971968-4a...	$75,000.00	$64,000.00
4	fab087e8-8c...	$83,000.00	$63,000.00

5. For each Branch get the list of the users that are not creators

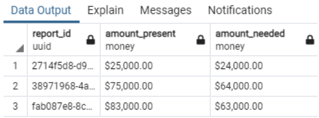
```
SELECT branch_id, first_name, last_name
FROM D1.user AS U FULL JOIN D1.report AS R
ON U.email = R.u_id
FULL JOIN D1.branch as B
ON U.b_id = B.branch_id
WHERE u_id IS NULL
```



	branch_id uuid	first_name character varying (200)	last_name character varying (200)
1	79cef980-631...	Nicol	Green
2	79cef980-631...	Lena	Smith
3	79cef980-631...	Paul	Miller
4	79cef980-631...	Alen	Lee
5	79cef980-631...	Tom	Taylor

6. Retrieve a list of project together with their report id, present budget amount needed which their present budget is more than amount needed. Also retrieve those which their amount needed is below 30000.

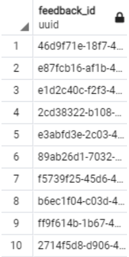
```
SELECT report_id, amount_present, amount_needed
FROM D1.report AS R INNER JOIN D1.budget as B
ON R.b_id = B.budget_id
WHERE amount_present > amount_needed
UNION
SELECT report_id, amount_present, amount_needed
FROM D1.report AS R INNER JOIN D1.budget as B
ON R.b_id = B.budget_id
WHERE amount_needed::numeric::int < 30000
```



	report_id uuid	amount_present money	amount_needed money
1	2714f5d8-d9...	$25,000.00	$24,000.00
2	38971968-4a...	$75,000.00	$64,000.00
3	fab087e8-8c...	$83,000.00	$63,000.00

7. Find the list of the Feedback_ID and the status_id that has the variable Current equal to True , also with the Report_ID that has the deadline before 20/02/2022.

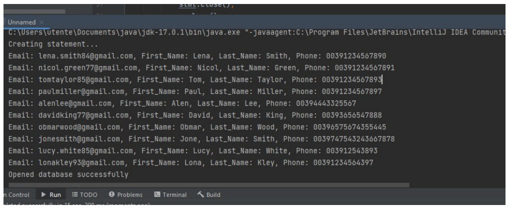
```
SELECT Feedback.Feedback_ID
FROM D1.Feedback
UNION
SELECT Status.Status_ID
FROM D1.Status
WHERE Status.Current_ = 'True'
UNION
SELECT Report.Report_ID
FROM D1.Report
WHERE Report.Deadline > '2022.02.20';
```



	feedback_id uuid
1	46d9f71e-18f7-4...
2	e87fcb16-af1b-4...
3	e1d2c40c-f2f3-4...
4	2cd38322-b108-...
5	e3abfd3e-2c03-4...
6	89ab26d1-7032-...
7	f5739f25-45d6-4...
8	b6ec1f04-c03d-4...
9	ff9f614b-1b67-4...
10	2714f5d8-d906-4...

JDBC Implementations of the Principal Queries and Visualization

```
public class App {
    //The JDBC driver to be used
    public static void main(String args[]) {
        Statement stmt = null;
        Connection c = null;
        try {
            Class.forName("org.postgresql.Driver");
            c = DriverManager
                    .getConnection("jdbc:postgresql://localhost:5432/postgres",
                            "postgres", "123456");
            System.out.println("Creating statement...");
            stmt = c.createStatement();
            String sql1;
            sql1 = "SELECT * FROM D1.User";
            ResultSet rs = stmt.executeQuery(sql1);

            while(rs.next()){
                String User_id  = rs.getString("Email");
                String Phone = rs.getString("Phone");
                String First_Name = rs.getString("First_Name");
                String Last_Name = rs.getString("Last_Name");

                System.out.print("Email: " + User_id);
                System.out.print(", First_Name: " + First_Name);
                System.out.print(", Last_Name: " + Last_Name);
                System.out.println(", Phone: " + Phone);
            }
            rs.close();
            stmt.close();
            c.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.err.println(e.getClass().getName()+": "+e.getMessage());
            System.exit(0);
        }
        System.out.println("Opened database successfully");
    }
}
```



```
C:\Users\utente\Documents\java\jdk-17.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communit
Creating statement...
Email: lena.smith84@gmail.com, First_Name: Lena, Last_Name: Smith, Phone: 00391234567890
Email: nicol.green77@gmail.com, First_Name: Nicol, Last_Name: Green, Phone: 00391234567891
Email: tomtaylor85@gmail.com, First_Name: Tom, Last_Name: Taylor, Phone: 00391234567893
Email: paulmiller@gmail.com, First_Name: Paul, Last_Name: Miller, Phone: 00391234567897
Email: alenlee@gmail.com, First_Name: Alen, Last_Name: Lee, Phone: 00394443325567
Email: davidking77@gmail.com, First_Name: David, Last_Name: King, Phone: 00393656547888
Email: obmarwood@gmail.com, First_Name: Obmar, Last_Name: Wood, Phone: 0039657567435445
Email: jonesmith@gmail.com, First_Name: Jone, Last_Name: Smith, Phone: 003974754324366787 8
Email: lucy.white85@gmail.com, First_Name: Lucy, Last_Name: White, Phone: 003912543893
Email: lonakley93@gmail.com, First_Name: Lona, Last_Name: Kley, Phone: 00391234564397
Opened database successfully
```

1

Team Contribution

- Alfi Baqiatus Shofi: Populate the Database: Example
- Xheina Pajaj: ER Schema and Relational Schema Update, Physical Schema, Principal Queries, JDBC Visualization
- Kader Çiçek: Populate the Database: Example
- Gauri Pravishi: Principal Queries, Jdbc Visualisation, Latex conversions.
- Hakan Ates: Relational Schema Update, Database Population, JDBC Visualization
- Mehmet Ozturk : Variations to the Relational Schema ,ER Schema and Relational Schema explanation
- Vanesa Ziu: ER Schema and Relational Schema Update, Physical Schema, Database Population, Principal Queries
- Mohammad Mehdi Zare: Principal Queries, Relational Schema Update, Cover
- Yekaterina Muradyan: Principal Queries Isidorus Mau Loko: Principal Queries