Campus administration

NAVRUZA TULKUNOVA

ALEKSEY YAKOVLEV

ROMAN TOKAREV

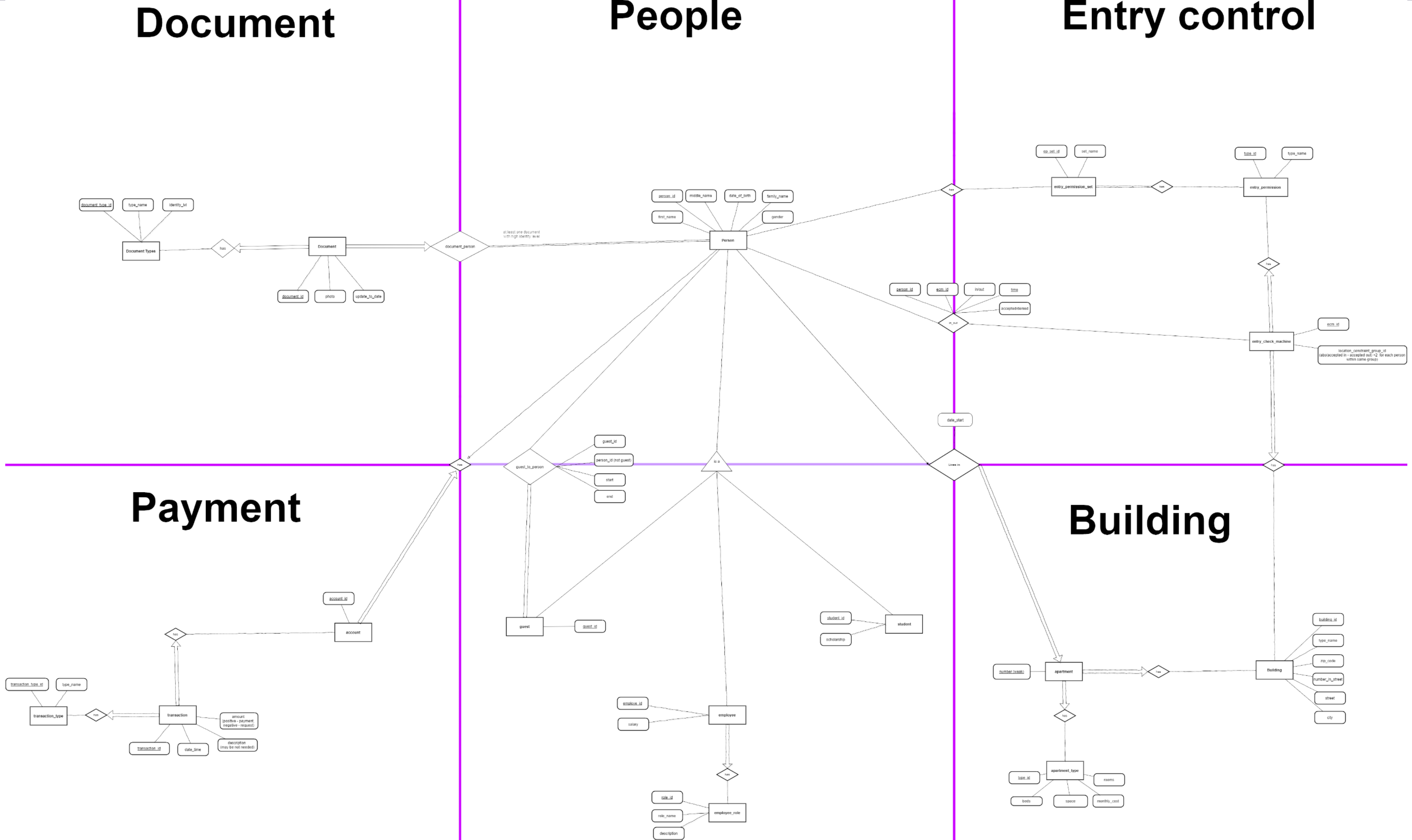
MO HAORAN

Idea of the Project

- Developing an application for managing all tasks related with the allocation and monitoring of the students’ housing facilities.
- It will make the dormitory administrator work easier, more efficient, and make fewer mistakes

Main features

- Apartments’ occupation control
- Guest control
- Students and personnel documents maintenance
- Payment control
- Personnel attendance control
- Access control

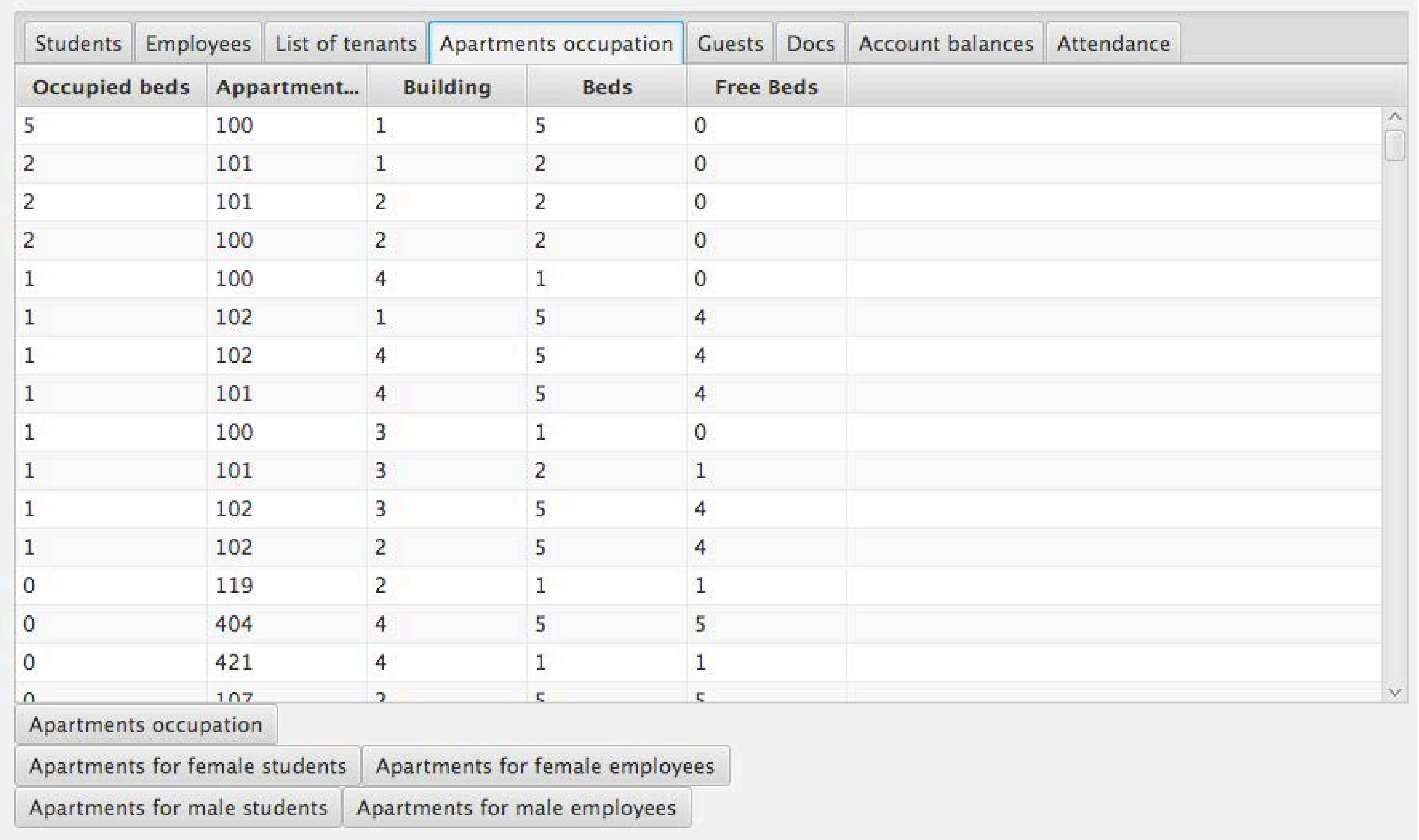

Document
People
Entry control
Payment
Building
document_type_id
type_name
identity_lvl
Document Types
has
Document
document_id
photo
update_to_date
document_person
at least one document
with high identity level
person_id
middle_name
date_of_birth
family_name
first_name
gender
Person
has
ep_set_id
set_name
entry_permission_set
has
type_id
type_name
entry_permission
has
person_id
ecm_id
in/out
time
accepted/denied
in_out
ecm_id
entry_check_machine
location_constraint_group_id
(abs(accepted in - accepted out) <2 for each person
within same group)
date_start
has
guest_to_person
guest_id
person_id (not guest)
start
end
is a
Lives in
has
account_id
account
has
transaction_type_id
type_name
transaction_type
has
transaction
amount
(positive - payment,
negative - request)
transaction_id
date_time
description
(may be not needed)
guest
guest_id
student_id
scholarship
student
employe_id
salary
employee
has
role_id
role_name
description
employee_role
building_id
type_name
zip_code
number_in_street
street
city
Building
number (weak)
apartment
has
has
apartment_type
type_id
rooms
beds
space
monthly_cost

Technologies

PostgreSQL

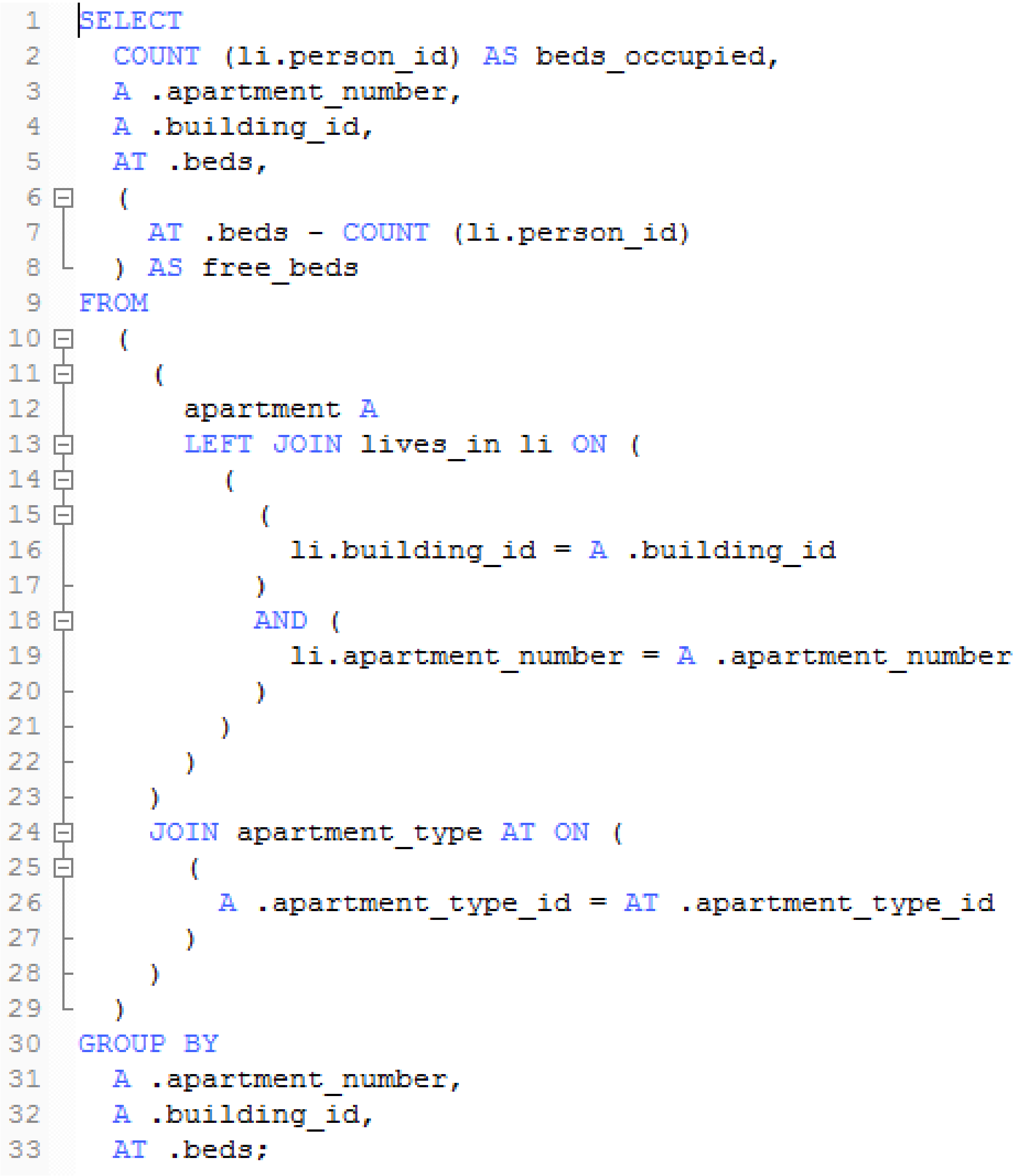
JavaFx

JDBC

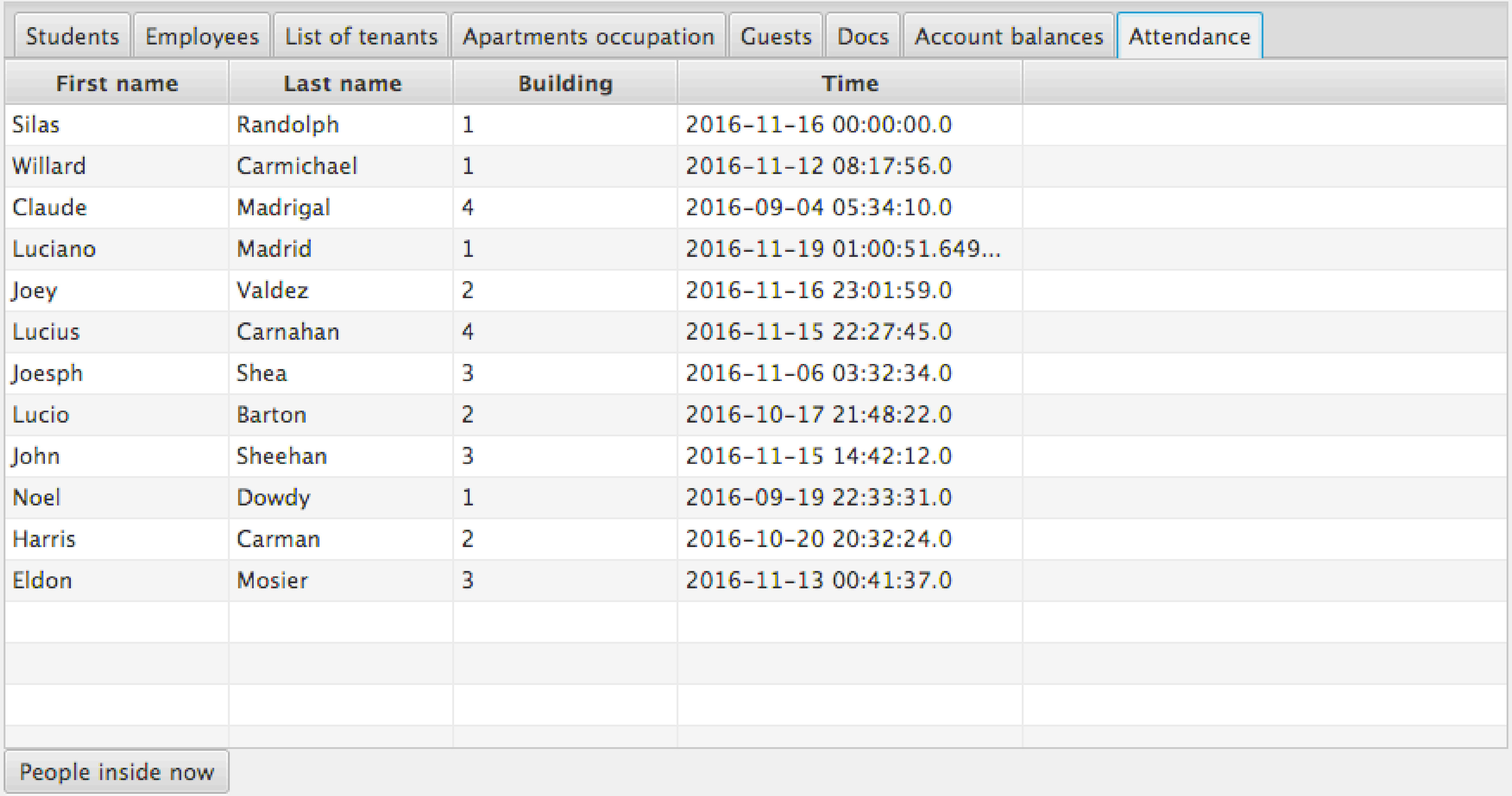

Students
Employees
List of tenants
Apartments occupation
Guests
Docs
Account balances
Attendance
Apartments occupation
Apartments for female students
Apartments for female employees
Apartments for male students
Apartments for male employees

Occupied beds	Appartment...	Building	Beds	Free Beds
5	100	1	5	0
2	101	1	2	0
2	101	2	2	0
2	100	2	2	0
1	100	4	1	0
1	102	1	5	4
1	102	4	5	4
1	101	4	5	4
1	100	3	1	0
1	101	3	2	1
1	102	3	5	4
1	102	2	5	4
0	119	2	1	1
0	404	4	5	5
0	421	4	1	1
0	107	2	5	5

```
SELECT * FROM apartment_occupation ORDER BY beds_occupied DESC
```

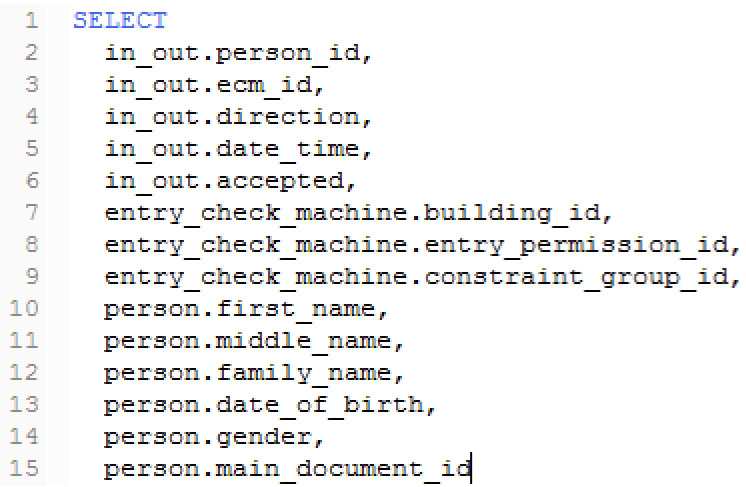


```sql
SELECT
  COUNT (li.person_id) AS beds_occupied,
  A .apartment_number,
  A .building_id,
  AT .beds,
  (
    AT .beds - COUNT (li.person_id)
  ) AS free_beds
FROM
  (
    (
      apartment A
      LEFT JOIN lives_in li ON (
        (
          (
            li.building_id = A .building_id
          )
          AND (
            li.apartment_number = A .apartment_number
          )
        )
      )
    )
    JOIN apartment_type AT ON (
      (
        A .apartment_type_id = AT .apartment_type_id
      )
    )
  )
GROUP BY
  A .apartment_number,
  A .building_id,
  AT .beds;
```

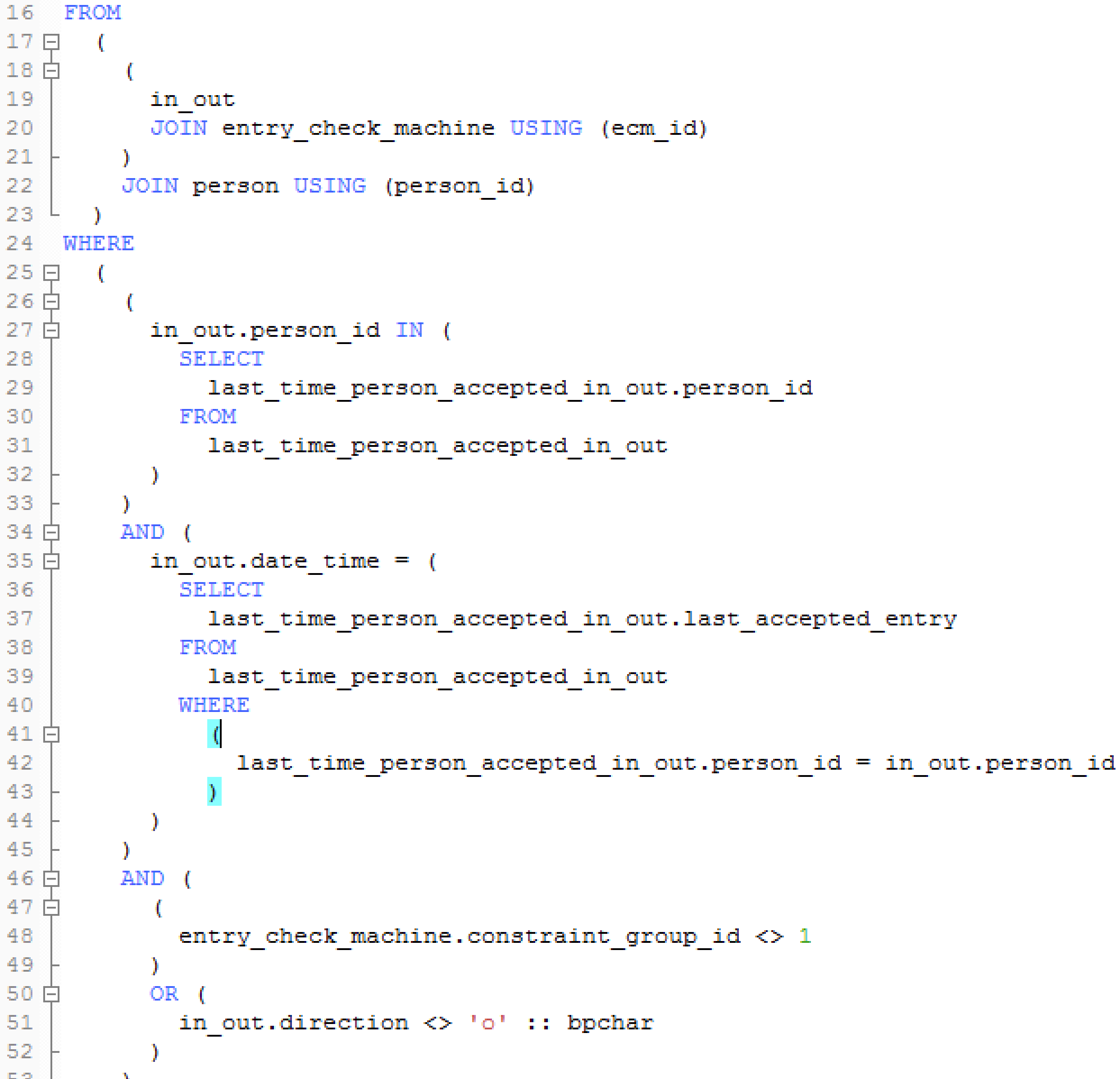


First name	Last name	Building	Time
Silas	Randolph	1	2016-11-16 00:00:00.0
Willard	Carmichael	1	2016-11-12 08:17:56.0
Claude	Madrigal	4	2016-09-04 05:34:10.0
Luciano	Madrid	1	2016-11-19 01:00:51.649...
Joey	Valdez	2	2016-11-16 23:01:59.0
Lucius	Carnahan	4	2016-11-15 22:27:45.0
Joesph	Shea	3	2016-11-06 03:32:34.0
Lucio	Barton	2	2016-10-17 21:48:22.0
John	Sheehan	3	2016-11-15 14:42:12.0
Noel	Dowdy	1	2016-09-19 22:33:31.0
Harris	Carman	2	2016-10-20 20:32:24.0
Eldon	Mosier	3	2016-11-13 00:41:37.0

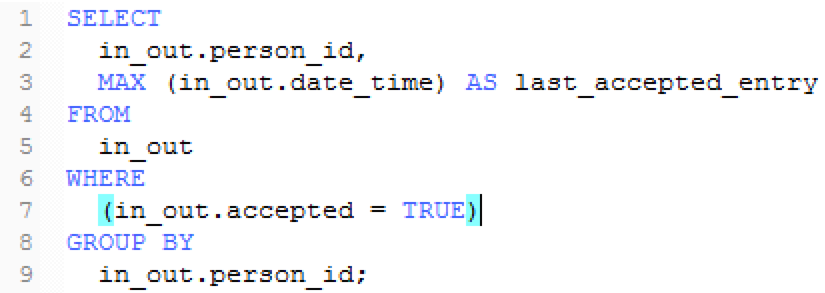
```
SELECT first_name, family_name, building_id, date_time FROM
persons_inside_campus_now
```



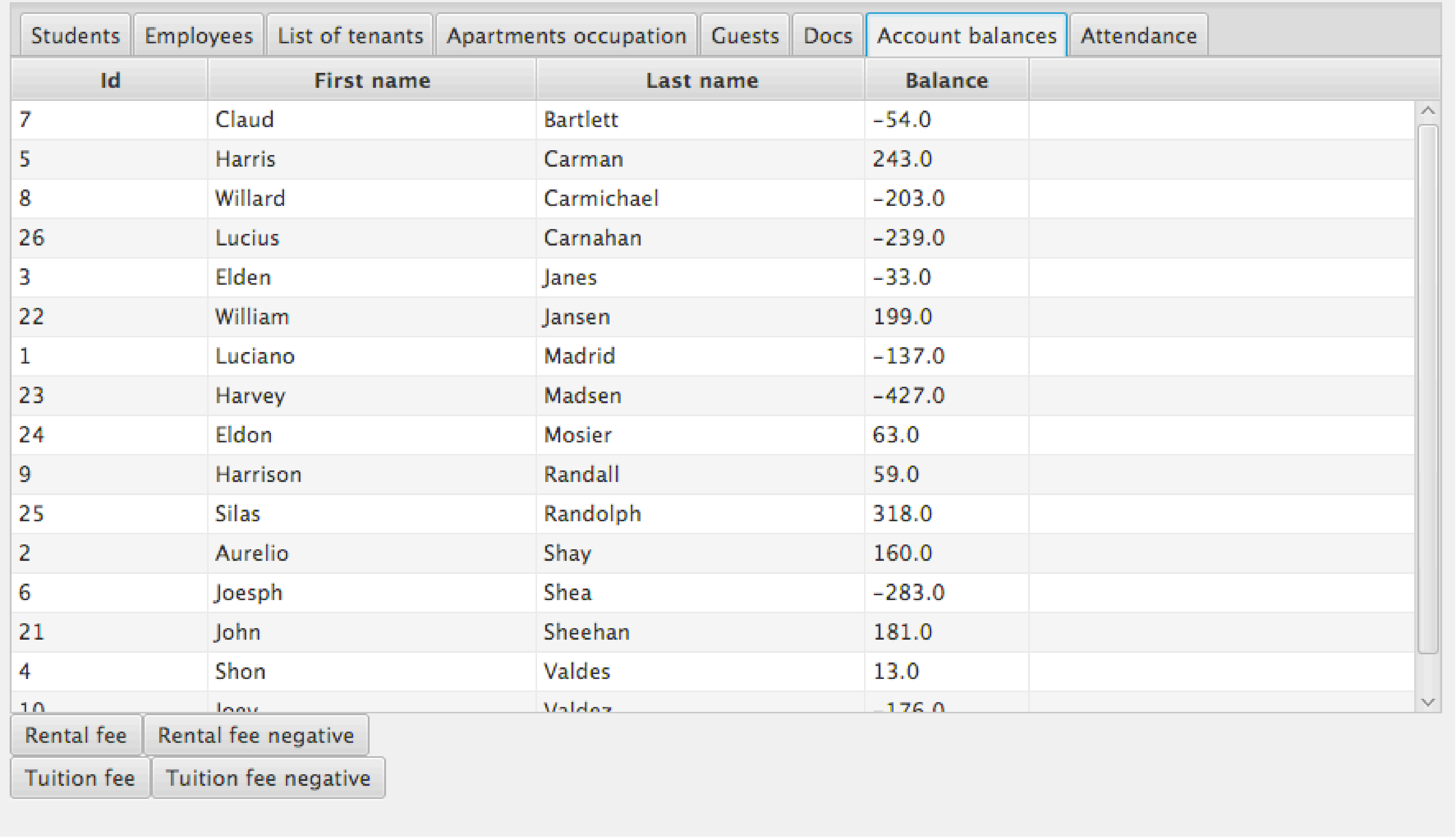
```
SELECT
  in_out.person_id,
  in_out.ecm_id,
  in_out.direction,
  in_out.date_time,
  in_out.accepted,
  entry_check_machine.building_id,
  entry_check_machine.entry_permission_id,
  entry_check_machine.constraint_group_id,
  person.first_name,
  person.middle_name,
  person.family_name,
  person.date_of_birth,
  person.gender,
  person.main_document_id
```



```
FROM
  (
    (
      in_out
      JOIN entry_check_machine USING (ecm_id)
    )
    JOIN person USING (person_id)
  )
WHERE
  (
    (
      in_out.person_id IN (
        SELECT
          last_time_person_accepted_in_out.person_id
        FROM
          last_time_person_accepted_in_out
      )
    )
    AND (
      in_out.date_time = (
        SELECT
          last_time_person_accepted_in_out.last_accepted_entry
        FROM
          last_time_person_accepted_in_out
        WHERE
          (
            last_time_person_accepted_in_out.person_id = in_out.person_id
          )
      )
    )
    AND (
      (
        entry_check_machine.constraint_group_id <> 1
      )
      OR (
        in_out.direction <> 'o' :: bpchar
      )
```

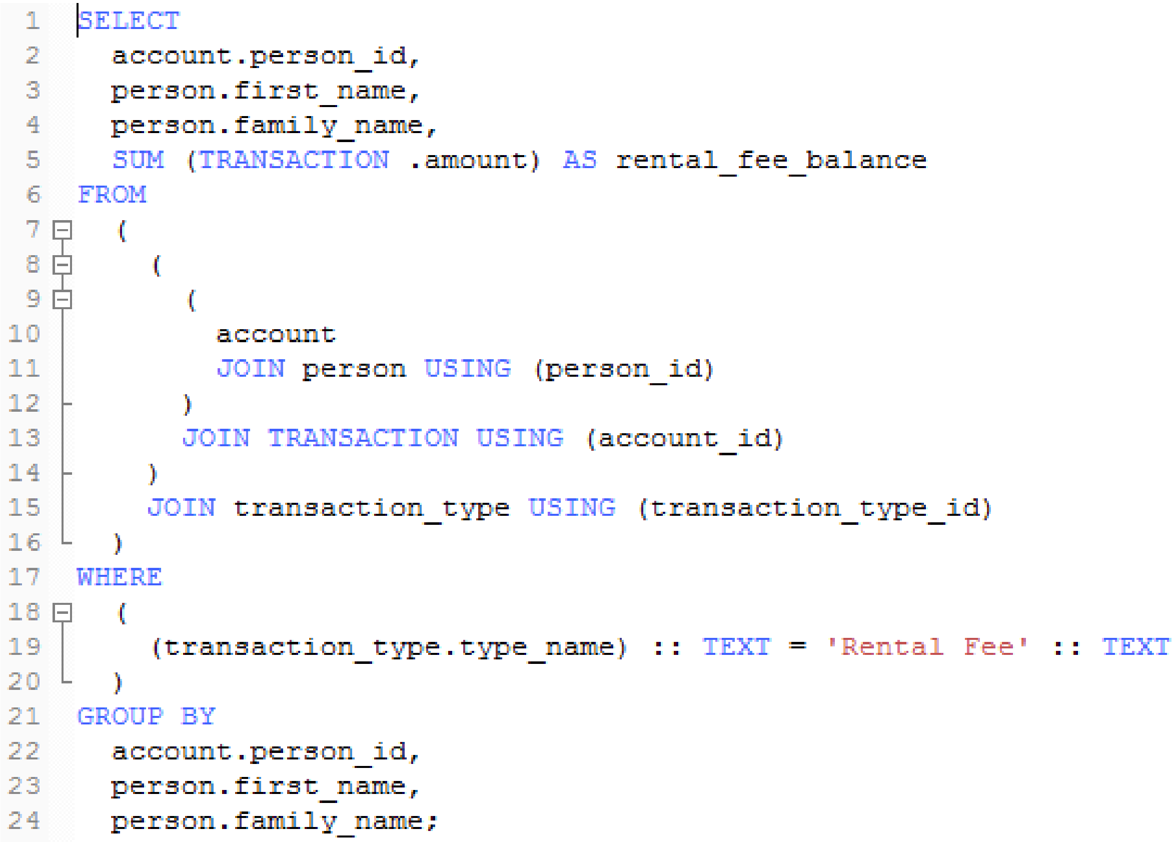


```
SELECT
  in_out.person_id,
  MAX (in_out.date_time) AS last_accepted_entry
FROM
  in_out
WHERE
  (in_out.accepted = TRUE)
GROUP BY
  in_out.person_id;
```

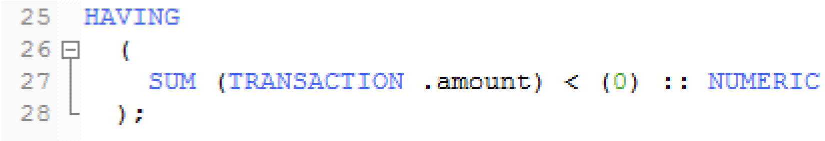


Id	First name	Last name	Balance
7	Claud	Bartlett	-54.0
5	Harris	Carman	243.0
8	Willard	Carmichael	-203.0
26	Lucius	Carnahan	-239.0
3	Elden	Janes	-33.0
22	William	Jansen	199.0
1	Luciano	Madrid	-137.0
23	Harvey	Madsen	-427.0
24	Eldon	Mosier	63.0
9	Harrison	Randall	59.0
25	Silas	Randolph	318.0
2	Aurelio	Shay	160.0
6	Joesph	Shea	-283.0
21	John	Sheehan	181.0
4	Shon	Valdes	13.0
10	Joey	Valdez	-176.0

```
SELECT * FROM rental_fee_balance ORDER BY family_name
```



```
SELECT
  account.person_id,
  person.first_name,
  person.family_name,
  SUM (TRANSACTION .amount) AS rental_fee_balance
FROM
  (
    (
      (
        account
        JOIN person USING (person_id)
      )
      JOIN TRANSACTION USING (account_id)
    )
    JOIN transaction_type USING (transaction_type_id)
  )
WHERE
  (
    (transaction_type.type_name) :: TEXT = 'Rental Fee' :: TEXT
  )
GROUP BY
  account.person_id,
  person.first_name,
  person.family_name;
```



```
HAVING
  (
    SUM (TRANSACTION .amount) < (0) :: NUMERIC
  );
```

Demonstration of the application