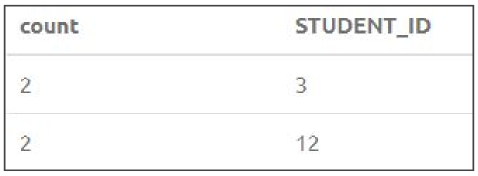
ZAŁOŻENIA BAZY DANYCH

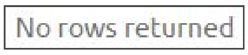
Relacja **STUDENT - CHARGE**:

1. Jeden uczeń może być przypisany do wielu opłat.
   select count("STUDENT_ID"), "STUDENT_ID" **from** "CHARGE"
   group by "STUDENT_ID" **having count**("STUDENT_ID") > 1



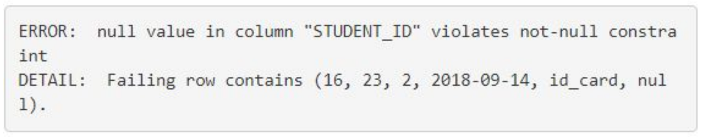
count	STUDENT_ID
2	3
2	12

2. Jedna opłata może być przypisana tylko do jednego ucznia.
   select count("CHARGE_ID"), "STUDENT_ID" **from** "CHARGE**" group by** "CHARGE_ID" **having count**("STUDENT_ID") > 1



```
No rows returned
```

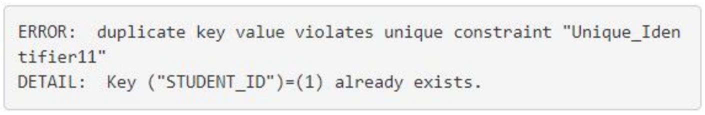
3. Nie można dodać opłaty bez połączenia jej z uczniem.
   insert into "CHARGE" ("CHARGE_ID", "AMOUNT_WITHOUT_INTEREST", "INTEREST", PAYMENT_DATE", "DESCRIPTION") **values** (16, 23, 2, '2018-09-14', 'id_card');



```
ERROR:  null value in column "STUDENT_ID" violates not-null constra
int
DETAIL:  Failing row contains (16, 23, 2, 2018-09-14, id_card, nul
l).
```

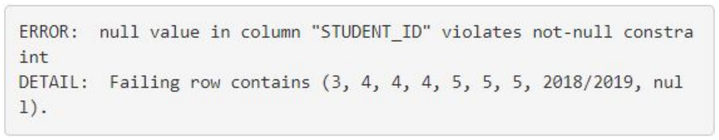
Relacja **STUDENT - FINAL_ASSESSMENT**:

1. Jeden uczeń może być przypisany do jednego kompletu ocen końcowych.
2. Jedna komplet ocen końcowych może być przypisany tylko do jednego ucznia.
   insert into "FINAL_ASSESSMENT" ("MATH", "CHEMISTRY", "HISTORY", "ENGLISH", "POLISH", "PHYSICS", "IT", "SCHOOL_YEAR", "STUDENT_ID") **values** (3, 4, 4, 4, 5, 5, 5, '2017/2018', 1);



```
ERROR:  duplicate key value violates unique constraint "Unique_Iden
tifier11"
DETAIL:  Key ("STUDENT_ID")=(1) already exists.
```

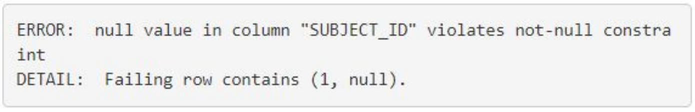
3. Nie można dodać ocen końcowych bez połączenia ich z uczniem.
   insert into "FINAL_ASSESSMENT" ("MATH", "CHEMISTRY", "HISTORY", "ENGLISH", "POLISH", "PHYSICS", "IT", "SCHOOL_YEAR") **values** (3, 4, 4, 4, 5, 5, 5, '2018/2019');



```
ERROR:  null value in column "STUDENT_ID" violates not-null constra
int
DETAIL:  Failing row contains (3, 4, 4, 4, 5, 5, 5, 2018/2019, nul
l).
```

Relacja **STUDENT - SUBJECT**:

1. Jeden uczeń może być przypisany do wielu przedmiotów.
2. Do jednego przedmiotu może być przypisanych wielu uczniów.
3. Nie można dodać ucznia bez połączenia go z przedmiotem.
   insert into "STUDENT_SUBJECT" ("STUDENT_ID") **values** (1);



```
ERROR:  null value in column "SUBJECT_ID" violates not-null constra
int
DETAIL:  Failing row contains (1, null).
```

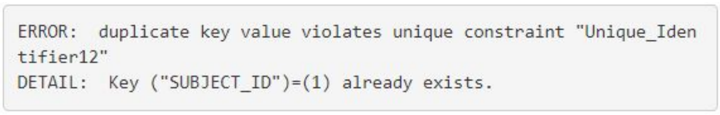
4. Nie można dodać przedmiotu bez połączenia go z uczniem.

Relacja **STUDENT - SPECIAL_INTERESTS_GROUP**:

1. Jeden uczeń może być przypisany do wielu grup zainteresowań.
2. Do jednej grupy zainteresowań może być przypisanych wielu uczniów.
3. Można dodać ucznia bez połączenia go z grupą zainteresowań.
4. Można dodać grupę zainteresowań bez połączenia jej z uczniem.

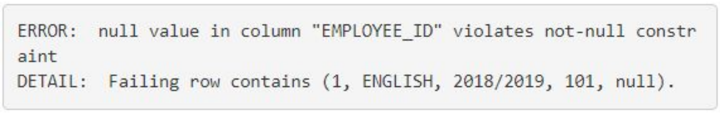
Relacja **TEACHER - SUBJECT**:

1. Jeden nauczyciel może być przypisany do wielu przedmiotów.
2. Do jednego przedmiotu może być przypisany jeden nauczyciel.
   insert into "SUBJECT" ("SUBJECT_ID", "SUBJECT_NAME", "SCHOOL_YEAR", "CLASSROOM_ID", "EMPLOYEE_ID") **values** (1, 'ENGLISH', '2018/2019', 101, 1);



```
ERROR:  duplicate key value violates unique constraint "Unique_Iden
tifier12"
DETAIL:  Key ("SUBJECT_ID")=(1) already exists.
```

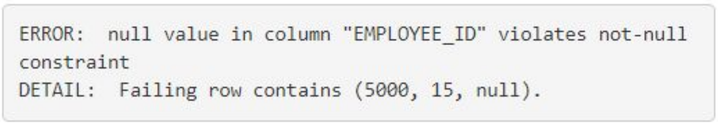
3. Można dodać nauczyciela bez połączenia go z przedmiotem.
4. Nie można dodać przedmiotu bez połączenia go z nauczycielem.
   insert into "SUBJECT" ("SUBJECT_ID", "SUBJECT_NAME", "SCHOOL_YEAR", "CLASSROOM_ID") **values** (1, 'ENGLISH', '2018/2019', 101);



```
ERROR:  null value in column "EMPLOYEE_ID" violates not-null constr
aint
DETAIL:  Failing row contains (1, ENGLISH, 2018/2019, 101, null).
```

Relacja **TEACHER - SPECIAL_INTERESTS_GROUP**:

1. Jeden nauczyciel może być przypisany do wielu grup zainteresowań.
2. Do jednej grupy zainteresowań może być przypisany jeden nauczyciel.
3. Nie można dodać grupy zainteresowań bez połączenia jej z nauczycielem.
   insert into "TEACHER_SIG" ("BUDGET", "SIG_ID") **values** (5000, 15);

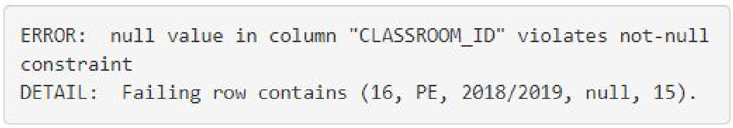


```
ERROR:  null value in column "EMPLOYEE_ID" violates not-null
constraint
DETAIL:  Failing row contains (5000, 15, null).
```

4. Można dodać nauczyciela bez połączenia go z grupą zainteresowań.

Relacja **SUBJECT - CLASSROOM**:

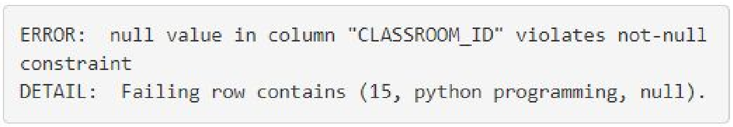
1. Jedna sala może być przypisana do wielu przedmiotów.
2. Jeden przedmiot może być przypisany do jednej sali.
3. Można dodać salę bez połączenia jej z przedmiotem.
4. Nie można dodać przedmiotu bez połączenia go z salą.
   insert into "SUBJECT" ("SUBJECT_ID", "SUBJECT_NAME", "SCHOOL_YEAR", "EMPLOYEE_ID") **values** (16, 'PE', '2018/2019', 15);



```
ERROR:  null value in column "CLASSROOM_ID" violates not-null
constraint
DETAIL:  Failing row contains (16, PE, 2018/2019, null, 15).
```

Relacja **SPECIAL_INTERESTS_GROUP - CLASSROOM**:

1. Jedna sala może być przypisana do wielu grup zainteresowań.
2. Jedna grupa zainteresowań może być przypisana do jednej sali.
3. Można dodać salę bez połączenia jej z grupą zainteresowań.
4. Nie można dodać grupy zainteresowań bez połączenia jej z salą.
   insert into "SPECIAL_INTERESTS_GROUP" ("SIG_ID", "SUBJECT_AREA") **values** (15, 'python programming');



```
ERROR:  null value in column "CLASSROOM_ID" violates not-null
constraint
DETAIL:  Failing row contains (15, python programming, null).
```

OPIS (ALGORYTM) KONWERSJI Z MODELU E-R

Krok 1: **Odwzorowanie zwykłych (silnych) zbiorów encji**.
Dla każdego zbioru encji w modelu E-R tworzona jest odpowiadająca mu relacjna w modelu relacyjnym. Utworzona relacja będzie zawierała wszystkie atrybuty proste zbioru encji, z którego powstaje.

Krok 2: **Odwzorowanie słabych zbiorów encji**.
Dla każdego słabego zbioru encji tworzona jest relacja zawierająca wszystkie proste atrybuty zbioru encji. Jednak w przypadku przedstawianego projektu nie mamy do czynienia z żadnym słabym zbiorem encji, dlatego ten krok pomijamy.

Krok 3: **Odwzorowanie binarnych związków 1:1.**
Opiera się na wykorzystaniu klucza obcego. Wybierana jest jedna z relacji odpowiadających jednemu ze zbiorów encji konkretnego związku i należy w niej umieścić dodatkowy atrybut (lub atrybuty) klucza obcego, który wskazuje na klucz głównej drugiej z relacji tego związku.

Krok 4: **Odwzorowanie binarnych związków 1:N.**
Należy wybrać relację będącą w związku po stronie N, a następnie wstawić do niej dodatkowy atrybut (lub atrybuty) klucza obcego, który wzkazuje na klucz gówny relacyjny będącej w związku po stronie 1.

Krok 5: **Odwzorowanie binarnych związków M:N.**
Należy stworzyć nową relację przeznaczoną dla tego związku. Nowo utworzona relacja powinna posiadać klucz obce, które wskazuje na klucze główne zbiorów encji będących w tym związku. Podobnie należy postąpić w przypadku związków posiadających własne atrybuty.

Krok 6: **Odwzorowanie atrybutów wielowartościowych.**
Dla każdego atrybutu wielowartościowego należy utworzyć nową relację. Nowo utworzona relacja będzie zawierać atrybuty odpowiadające wielowartośiowemu atrybutowi zbioru encji i atrybutom klucza głownego relacji reprezentującej zbiór encji posiadający ten atrybut wielowartościowy.

Krok 7: **Odwzorowanie dziedziczenia.**
W przypadku przedstawianego projektu mamy do czynienia z dziedzieczeniem obowiązkowym, rozłącznym. Oznacza to, że dla każdej z podklas należy utworzyć nową relację, która będzie zawierać wszystkie atrybuty nadklasy oraz atrybuty związane wyłącznie z daną podklasą, jeśli takie istnieją.

DOPISYWANIE DANYCH DO BAZY DANYCH

DOPISYWANIE DO TABLICY **'STUDENT'**:

INSERT INTO "STUDENT" ("STUDENT_ID", "SURNAME", "NAME", "DATE_OF_BIRTH", "FORM", "STIPEND") values [...]
--- STUDENT_ID (PK) IS NOT NULL ---> BIGINT
--- SURNAME IS NOT NULL ---> TEXT
--- NAME IS NOT NULL ---> TEST
--- DATE_OF_BIRTH IS NOT NULL ---> DATE
--- FORM IS NOT NULL ---> TEXT
--- STIPEND ---> BOOLEAN

<-->

DOPISYWANIE DO TABLICY **'FINAL_ASSESSMENT'**:

INSERT INTO "FINAL_ASSESSMENT" ("MATH", "CHEMISTRY", "HISTORY", "ENGLISH", "POLISH", "PHYSICS", "IT", "SCHOOL_YEAR", "STUDENT_ID") values [...]
--- MATH DEFAULT 0 ---> BIGINT
--- CHEMISTRY DEFAULT 0 ---> BIGINT
--- HISTORY DEFAULT 0 ---> BIGINT
--- ENGLISH DEFAULT 0 ---> BIGINT
--- POLISH DEFAULT 0 ---> BIGINT
--- PHYSICS DEFAULT 0 ---> BIGINT
--- IT DEFAULT 0 ---> BIGINT
--- SCHOOL_YEAR IS NOT NULL ---> TEXT
--- STUDENT_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'STUDENT_ADDRESS'**:

INSERT INTO "STUDENT_ADDRESS" ("LOCALITY", "POSTCODE", "STREET", "NO", "STUDENT_ID") VALUES [...]
--- LOCALITY IS NOT NULL ---> TEXT
--- POSTCODE IS NOT NULL ---> TEXT
--- STREET IS NOT NULL ---> TEXT
--- NO IS NOT NULL ---> TEXT
--- STUDENT_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'CHARGE'**:
insert into "CHARGE" ("CHARGE_ID", "AMOUNT_WITHOUT_INTEREST", "INTEREST", "PAYMENT_DATE", "DESCRIPTION", "STUDENT_ID") values [...]
--- CHARGE_ID (PK) IS NOT NULL ---> BIGINT
--- AMOUNT_WITHOUT_INTEREST IS NOT NULL ---> BIGINT
--- INSTEREST IS NOT NULL ---> BIGINT
--- PAYMENT_DATE IS NOT NULL ---> DATE
--- DESRIPTION IS NOT NULL ---> TEXT
--- STUDNET_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'TEACHER'**:

INSERT INTO "TEACHER" ("EMPLOYEE_ID", "TEACHER_ID", "SURNAME", "NAME", "DATE_OF_BIRTH", "SALARY", "CAREER_LADDER", "EMPLOYEE_SUPERVISOR_ID") values [...]
--- EMPLOYEE_ID (PK) IS NOT NULL ---> BIGINT
--- TEACHER_ID IS NOT NULL UNIQUE ---> BIGINT
--- SURNAME IS NOT NULL ---> TEXT
--- NAME IS NOT NULL ---> TEXT
--- DATE_OF_BIRTH IS NOT NULL ---> DATE
--- SALARY IS NOT NULL ---> BIGINT
--- CAREER_LADDER IS NOT NULL ---> TEXT
--- EMPLOYEE_SUPERVISOR_ID ---> BIGINT
<-->

DOPISYWANIE DO TABLICY **'TRAINING_COMPLETED'**:

INSERT INTO "TRAINING_COMPLETED" ("TRAINING_NAME", "TRAINING_DATE") values [...]
--- TRAINING_NAME IS NOT NULL ---> TEXT
--- TRAINING_DATE IS NOT NULL ---> DATE
<-->

DOPISYWANIE DO TABLICY **'TEACHER_TRAINING'**:

INSERT INTO "TEACHER_COMPLETED" ("TRAINING_NAME", "EMPLOYEE_ID") values [...]
--- TRAINING_NAME (FK) IS NOT NULL ---> TEXT
--- EMPLOYEE_IDIS (FK) NOT NULL ---> BIGINT
<-->

DOPISYWANIE DO TABLICY **'ADMINISTRATION'**:

INSERT INTO "ADMINISTRATION" ("EMPLOYEE_ID", "ADMINISTRATION_ID", "SURNAME", "NAME", "DATE_OF_BIRTH", "SALARY", "POSITION") values [...]
--- EMPLOYEE_ID (PK) IS NOT NULL ---> BIGINT
--- ADMINISTRATION_ID IS NOT NULL UNIQUE ---> BIGINT
--- SURNAME IS NOT NULL ---> TEXT
--- NAME IS NOT NULL ---> TEXT
--- DATE_OF_BIRTH IS NOT NULL ---> DATE
--- SALARY IS NOT NULL ---> BIGINT
--- POSITION IS NOT NULL ---> TEXT
<--->

DOPISYWANIE DO TABLICY **'MAINTENANCE'**:

INSERT INTO "MAINTENANCE" ("EMPLOYEE_ID", "MAINTENANCE_ID", "SURNAME", "NAME", "DATE_OF_BIRTH", "SALARY", "POSITION") values [...]
--- EMPLOYEE_ID (PK) IS NOT NULL ---> BIGINT
--- MAINTENANCE_ID IS NOT NULL UNIQUE ---> BIGINT
--- SURNAME IS NOT NULL ---> TEXT
--- NAME IS NOT NULL ---> TEXT
--- DATE_OF_BIRTH IS NOT NULL ---> DATE
--- SALARY IS NOT NULL ---> BIGINT
--- POSITION IS NOT NULL ---> TEXT
<--->

DOPISYWANIE DO TABLICY
'TEACHER_ADDRESS'/**'ADMINISTRATION_ADDRESS'**/**'MAINTENANCE_ADDRESS'**:

INSERT INTO "MAINTENANCE_ADDRESS" ("LOCALITY", "POSTCODE", "STREET", "NO", "MAINTENANCE_ID") VALUES [...]
--- LOCALITY IS NOT NULL ---> TEXT
--- POSTCODE IS NOT NULL ---> TEXT
--- STREET IS NOT NULL ---> TEXT
--- NO IS NOT NULL ---> TEXT
--- EMPLOYEE_ID (FK) IS NOT NULL ---> BIGINT

<--->

DOPISYWANIE DO TABLICY **'CLASSROOM'**:

INSERT INTO "CLASSROOM" ("CLASSROOM_ID", "COMPLEX") values [...]
--- CLASSROOM (PK) IS NOT NULL ---> BIGINT
--- COMPLEX IS NOT NULL ---> BIGINT

<--->

DOPISYWANIE DO TABLICY **'SUBJECT'**:

INSERT INTO "CLASSROOM" ("SUBJECT_ID", "SUBJECT_NAME", "SCHOOL_YEAR", "CLASSROOM_ID", "EMPLOYEE_ID") values [...]
--- SUBJECT_ID (PK) IS NOT NULL ---> BIGINT
--- SUBJECT_NAME IS NOT NULL ---> TEXT
--- SCHOOL_YEAR IS NOT NULL ---> TEXT
--- CLASSROOM_ID (FK) IS NOT NULL ---> BIGINT
--- EMPLOYEE_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'STUDENT_SUBJECT'**:

INSERT INTO "STUDENT_CLASSROOM" ("STUDENT_ID", "SUBJECT_ID") values [...]
--- STUDENT_ID (FK) ---> BIGINT
--- SUBJECT_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'SPECIAL_INTERESTS_GROUP'**:

INSERT INTO "SPECIAL_INTERESTS_GROUP" ("SIG_ID", "SUBJECT_AREA", "CLASSROOM_ID") values [...]
--- SIG_ID (PK) IS NOT NULL ---> BIGINT
--- SUBJECT_AREA IS NOT NULL ---> TEXT
--- CLASSROOM_ID (FK) IS NOT NULL ---> BIGINT

<-->

DOPISYWANIE DO TABLICY **'TEACHER_SIG'**:

INSERT INTO "TEACHER_SIG" ("BUDGET", "SIG_ID", "EMPLOYEE_ID") values [...]
--- BUDGET IS NOT NULL ---> BIGINT
--- SIG_ID (FK) ---> BIGINT
--- EMPLOYEE_ID (FK) IS NOT NULL ---> BIGINT

<-->

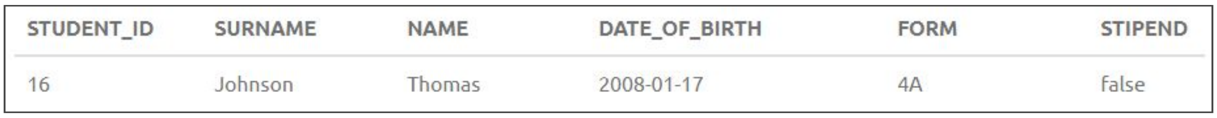
DOPISYWANIE DO TABLICY **'STUDENT_SIG'**:

INSERT INTO "STUDENT_SIG" ("STUDENT_ID", "SIG_ID") values [...]
--- STUDENT_ID (FK) ---> BIGINT
--- SIG_ID (FK) ---> BIGINT
<-->

MODYFIKOWANIE DANYCH

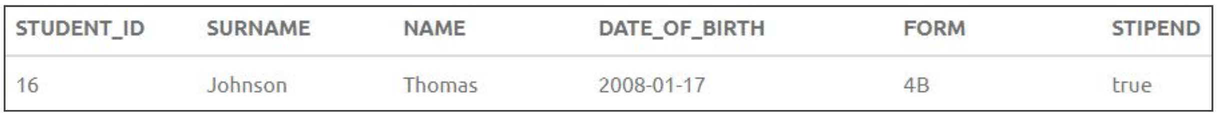
Dane należy modyfikować zgodnie ze schematem:
update NAZWA_TABELI
set NAZWA_KOLUMNY = WARTOSC, NAZWA_KOLUMNY_2 = WARTOSC2, ...
where WARUNEK

Modyfikowania danych na przykładzie tabeli “STUDENT”:



STUDENT_ID	SURNAME	NAME	DATE_OF_BIRTH	FORM	STIPEND
16	Johnson	Thomas	2008-01-17	4A	false

update "STUDENT" **set** "FORM" = '4B', "STIPEND" = true **where** "STUDENT_ID" = 16

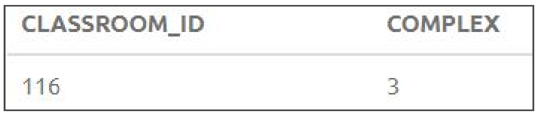


STUDENT_ID	SURNAME	NAME	DATE_OF_BIRTH	FORM	STIPEND
16	Johnson	Thomas	2008-01-17	4B	true

USUWANIE DANYCH

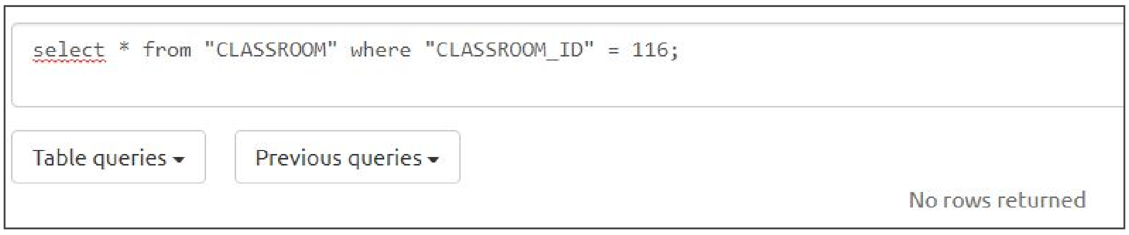
Dane należy usuwać zgodnie ze schematem:
delete from NAZWA_TABELI
where WARUNEK

Usuwanie danych na przykładzie tabeli “CLASSROOM”:



CLASSROOM_ID	COMPLEX
116	3

delete from "CLASSROOM" **where** "CLASSROOM_ID" = 116;

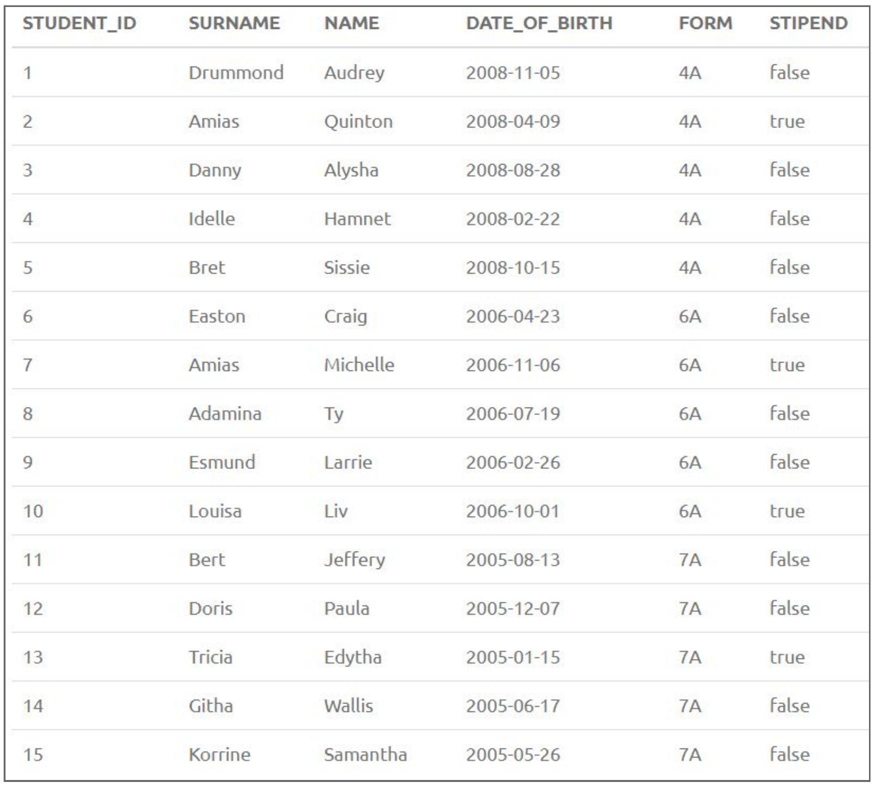


```
select * from "CLASSROOM" where "CLASSROOM_ID" = 116;
```

No rows returned

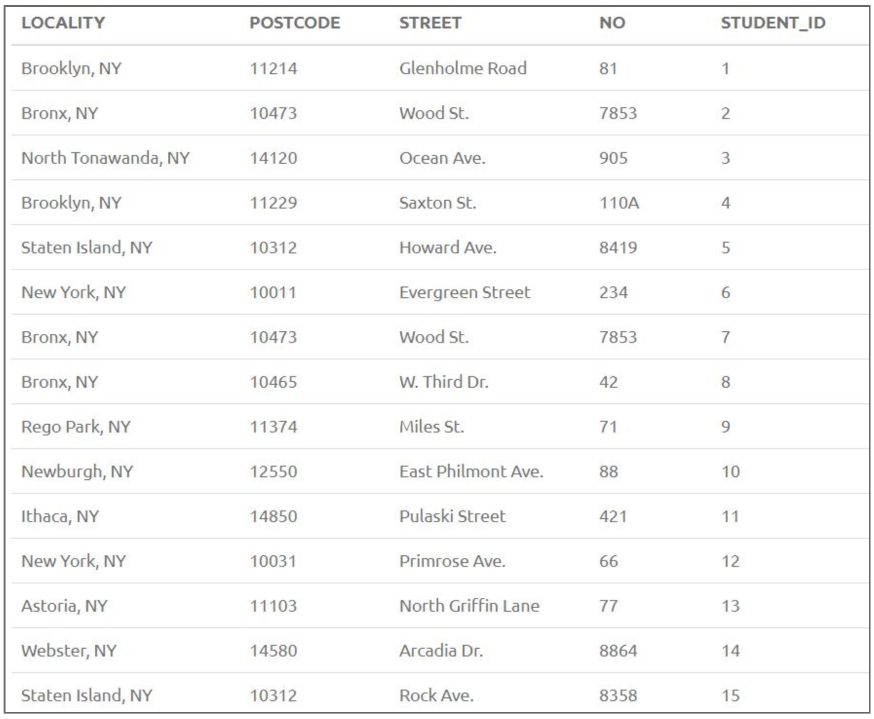
WYDRUKI TABEL

1. Tabela **STUDENT** zawiera informacje o uczniach placówki.



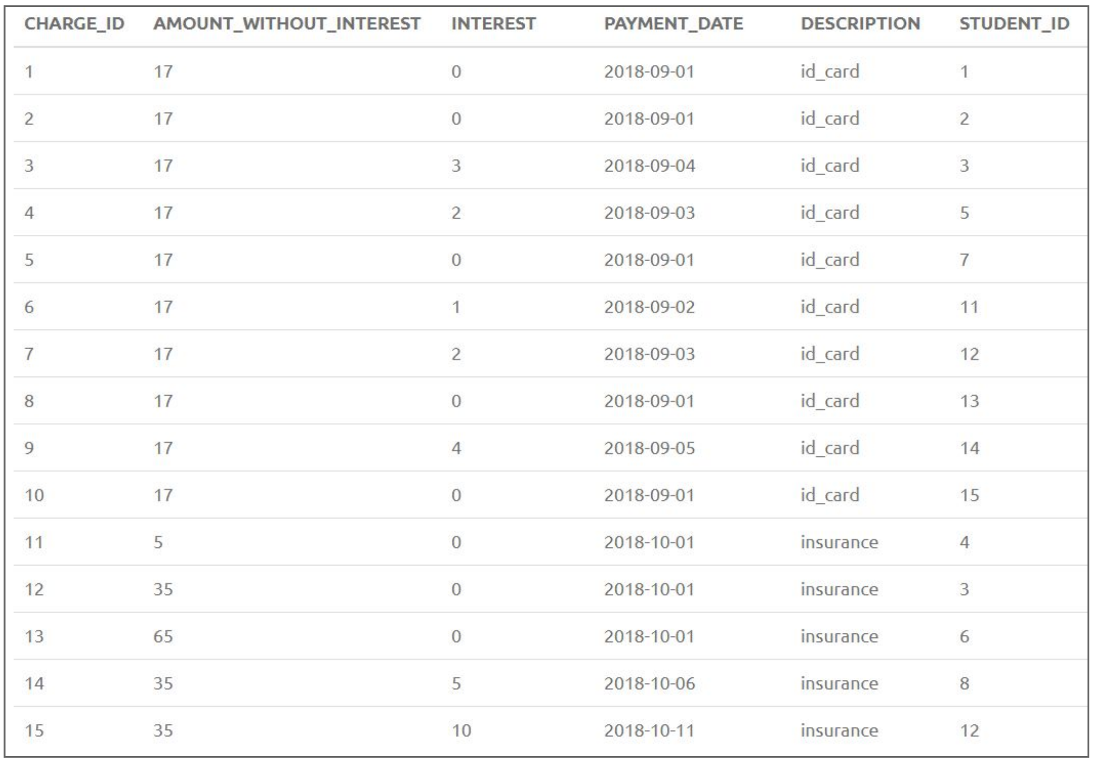
STUDENT_ID	SURNAME	NAME	DATE_OF_BIRTH	FORM	STIPEND
1	Drummond	Audrey	2008-11-05	4A	false
2	Amias	Quinton	2008-04-09	4A	true
3	Danny	Alysha	2008-08-28	4A	false
4	Idelle	Hamnet	2008-02-22	4A	false
5	Bret	Sissie	2008-10-15	4A	false
6	Easton	Craig	2006-04-23	6A	false
7	Amias	Michelle	2006-11-06	6A	true
8	Adamina	Ty	2006-07-19	6A	false
9	Esmund	Larrie	2006-02-26	6A	false
10	Louisa	Liv	2006-10-01	6A	true
11	Bert	Jeffery	2005-08-13	7A	false
12	Doris	Paula	2005-12-07	7A	false
13	Tricia	Edytha	2005-01-15	7A	true
14	Githa	Wallis	2005-06-17	7A	false
15	Korrine	Samantha	2005-05-26	7A	false

2. Tabela **STUDENT_ADDRESS** zawiera adresy korespondencyjne uczniów placówki.



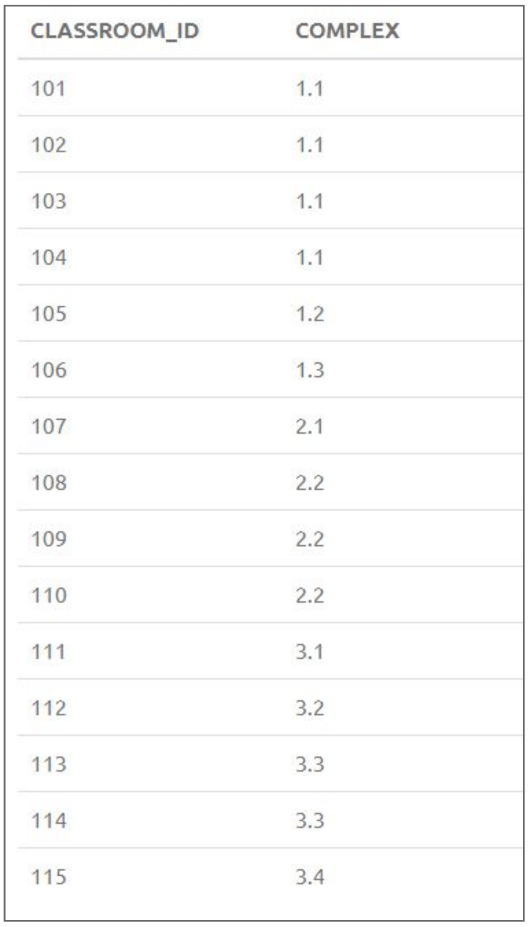
LOCALITY	POSTCODE	STREET	NO	STUDENT_ID
Brooklyn, NY	11214	Glenholme Road	81	1
Bronx, NY	10473	Wood St.	7853	2
North Tonawanda, NY	14120	Ocean Ave.	905	3
Brooklyn, NY	11229	Saxton St.	110A	4
Staten Island, NY	10312	Howard Ave.	8419	5
New York, NY	10011	Evergreen Street	234	6
Bronx, NY	10473	Wood St.	7853	7
Bronx, NY	10465	W. Third Dr.	42	8
Rego Park, NY	11374	Miles St.	71	9
Newburgh, NY	12550	East Philmont Ave.	88	10
Ithaca, NY	14850	Pulaski Street	421	11
New York, NY	10031	Primrose Ave.	66	12
Astoria, NY	11103	North Griffin Lane	77	13
Webster, NY	14580	Arcadia Dr.	8864	14
Staten Island, NY	10312	Rock Ave.	8358	15

3. Tabela **CHARGE** zawiera zaksięgowane opłaty dokonane przez uczniów placówki.



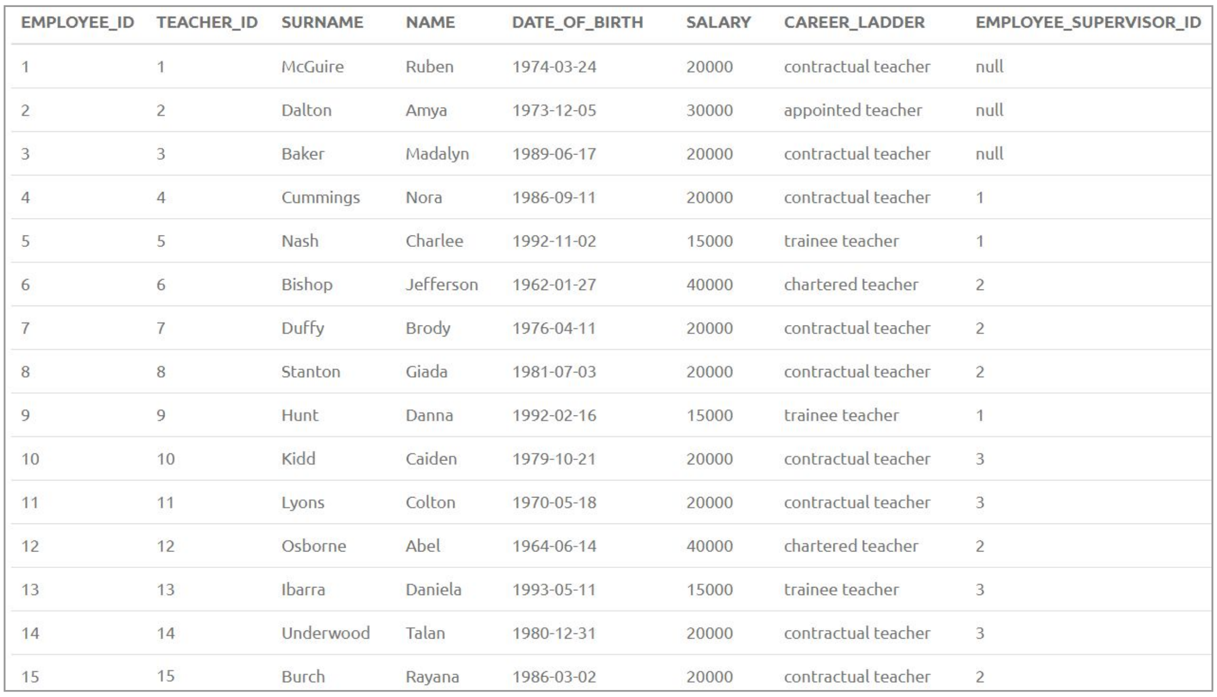
CHARGE_ID	AMOUNT_WITHOUT_INTEREST	INTEREST	PAYMENT_DATE	DESCRIPTION	STUDENT_ID
1	17	0	2018-09-01	id_card	1
2	17	0	2018-09-01	id_card	2
3	17	3	2018-09-04	id_card	3
4	17	2	2018-09-03	id_card	5
5	17	0	2018-09-01	id_card	7
6	17	1	2018-09-02	id_card	11
7	17	2	2018-09-03	id_card	12
8	17	0	2018-09-01	id_card	13
9	17	4	2018-09-05	id_card	14
10	17	0	2018-09-01	id_card	15
11	5	0	2018-10-01	insurance	4
12	35	0	2018-10-01	insurance	3
13	65	0	2018-10-01	insurance	6
14	35	5	2018-10-06	insurance	8
15	35	10	2018-10-11	insurance	12

4. Tabela **CLASSROOM** zawiera informacje dotyczące sal dydaktycznych w placówce.



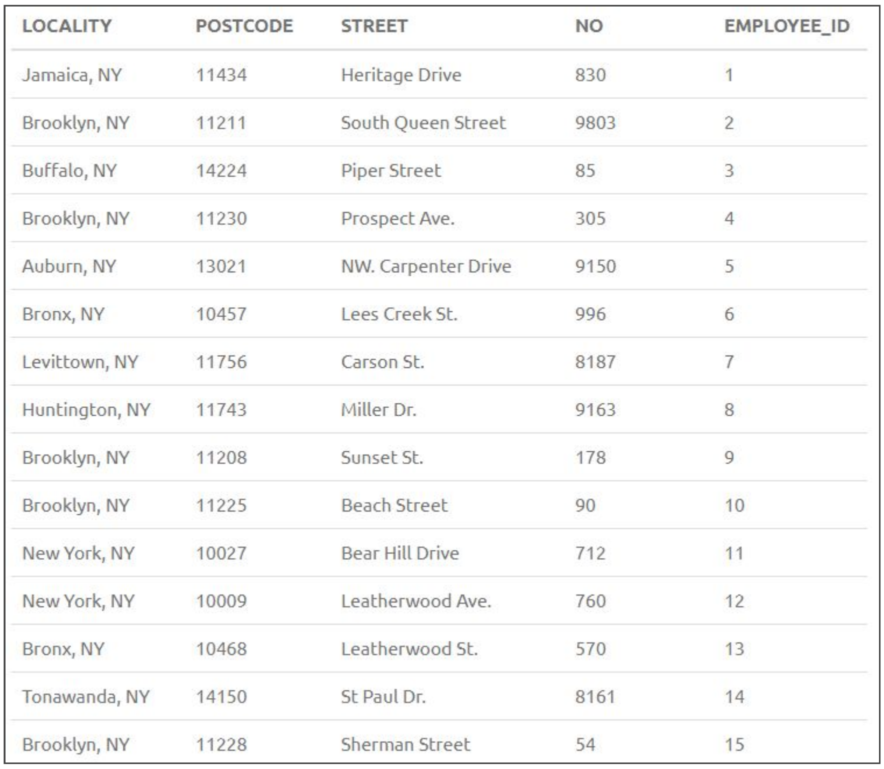
CLASSROOM_ID	COMPLEX
101	1.1
102	1.1
103	1.1
104	1.1
105	1.2
106	1.3
107	2.1
108	2.2
109	2.2
110	2.2
111	3.1
112	3.2
113	3.3
114	3.3
115	3.4

5. Tabela **TEACHER** zawiera informacje o pracownikach dydaktycznych placówki.



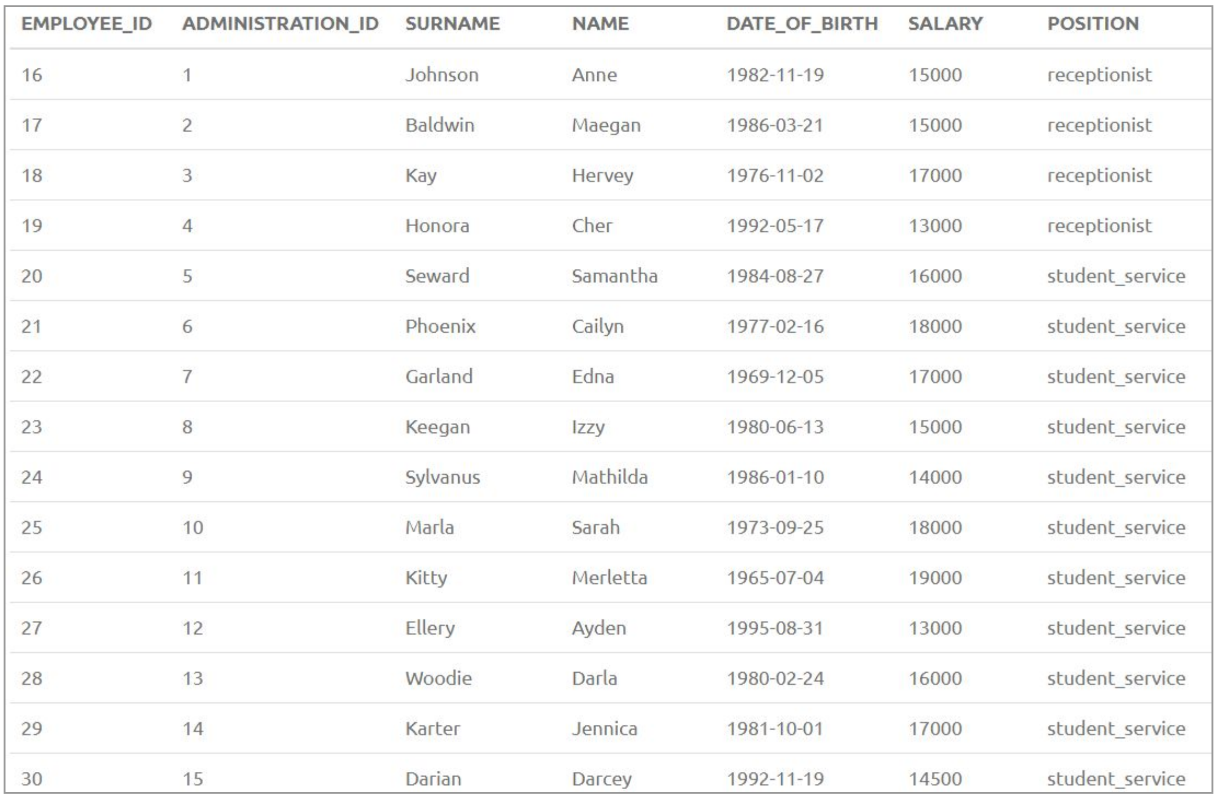
EMPLOYEE_ID	TEACHER_ID	SURNAME	NAME	DATE_OF_BIRTH	SALARY	CAREER_LADDER	EMPLOYEE_SUPERVISOR_ID
1	1	McGuire	Ruben	1974-03-24	20000	contractual teacher	null
2	2	Dalton	Amya	1973-12-05	30000	appointed teacher	null
3	3	Baker	Madalyn	1989-06-17	20000	contractual teacher	null
4	4	Cummings	Nora	1986-09-11	20000	contractual teacher	1
5	5	Nash	Charlee	1992-11-02	15000	trainee teacher	1
6	6	Bishop	Jefferson	1962-01-27	40000	chartered teacher	2
7	7	Duffy	Brody	1976-04-11	20000	contractual teacher	2
8	8	Stanton	Giada	1981-07-03	20000	contractual teacher	2
9	9	Hunt	Danna	1992-02-16	15000	trainee teacher	1
10	10	Kidd	Caiden	1979-10-21	20000	contractual teacher	3
11	11	Lyons	Colton	1970-05-18	20000	contractual teacher	3
12	12	Osborne	Abel	1964-06-14	40000	chartered teacher	2
13	13	Ibarra	Daniela	1993-05-11	15000	trainee teacher	3
14	14	Underwood	Talan	1980-12-31	20000	contractual teacher	3
15	15	Burch	Rayana	1986-03-02	20000	contractual teacher	2

6. Tabela **TEACHER_ADDRESS** zawiera adresy korespondencyjne pracowników dydaktycznych placówki.



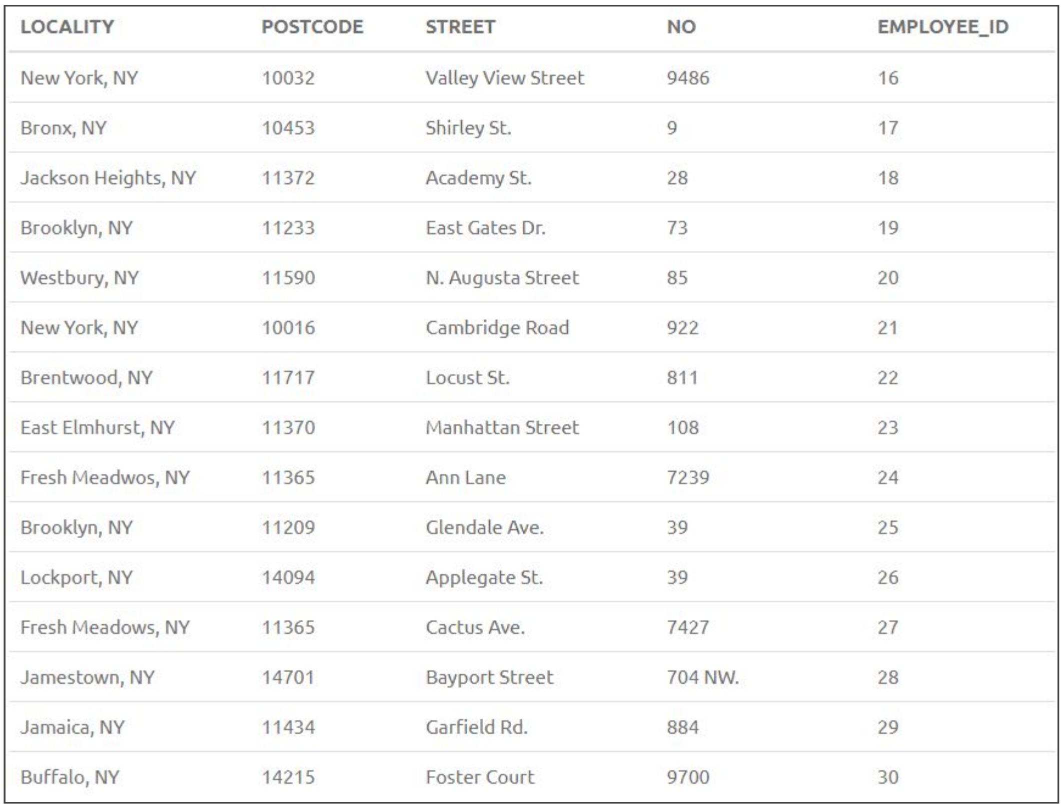
LOCALITY	POSTCODE	STREET	NO	EMPLOYEE_ID
Jamaica, NY	11434	Heritage Drive	830	1
Brooklyn, NY	11211	South Queen Street	9803	2
Buffalo, NY	14224	Piper Street	85	3
Brooklyn, NY	11230	Prospect Ave.	305	4
Auburn, NY	13021	NW. Carpenter Drive	9150	5
Bronx, NY	10457	Lees Creek St.	996	6
Levittown, NY	11756	Carson St.	8187	7
Huntington, NY	11743	Miller Dr.	9163	8
Brooklyn, NY	11208	Sunset St.	178	9
Brooklyn, NY	11225	Beach Street	90	10
New York, NY	10027	Bear Hill Drive	712	11
New York, NY	10009	Leatherwood Ave.	760	12
Bronx, NY	10468	Leatherwood St.	570	13
Tonawanda, NY	14150	St Paul Dr.	8161	14
Brooklyn, NY	11228	Sherman Street	54	15

7. Tabela **ADMINISTRATION** zawiera informacje o pracownikach administracji placówki.



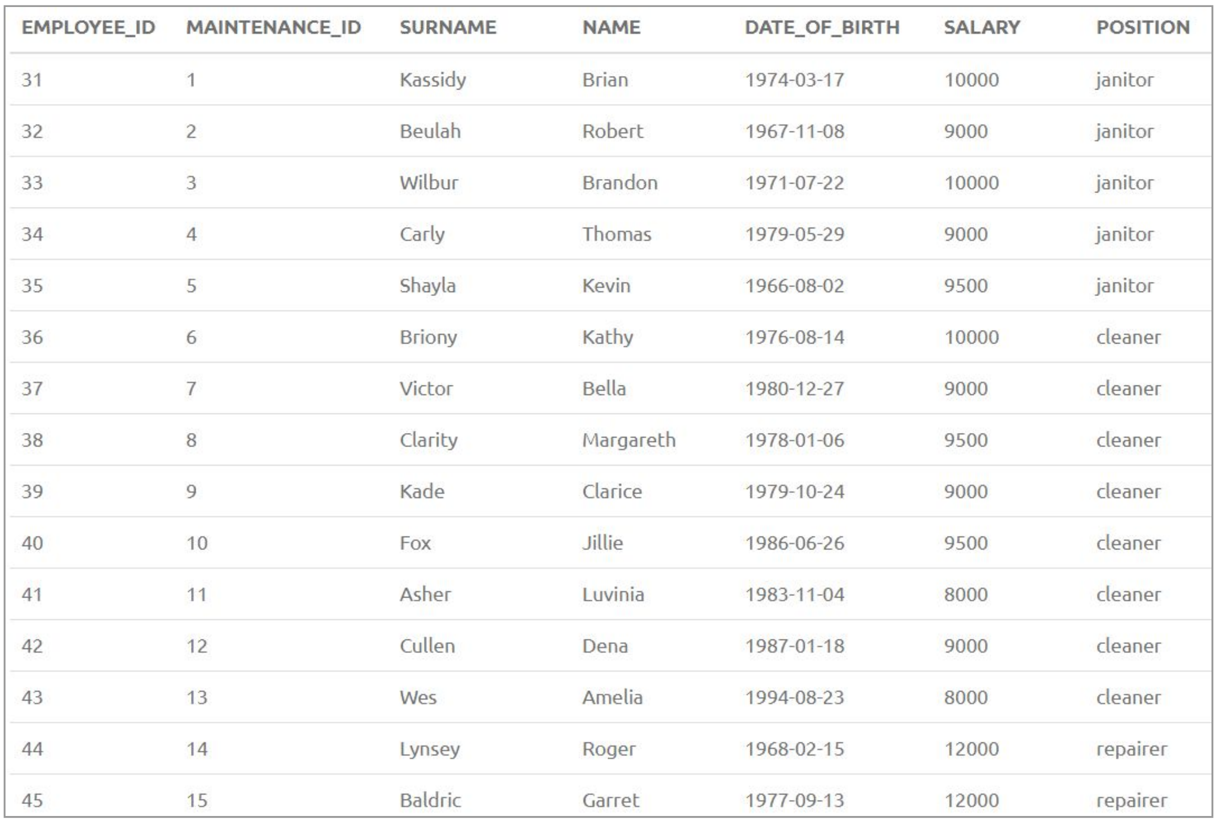
EMPLOYEE_ID	ADMINISTRATION_ID	SURNAME	NAME	DATE_OF_BIRTH	SALARY	POSITION
16	1	Johnson	Anne	1982-11-19	15000	receptionist
17	2	Baldwin	Maegan	1986-03-21	15000	receptionist
18	3	Kay	Hervey	1976-11-02	17000	receptionist
19	4	Honora	Cher	1992-05-17	13000	receptionist
20	5	Seward	Samantha	1984-08-27	16000	student_service
21	6	Phoenix	Cailyn	1977-02-16	18000	student_service
22	7	Garland	Edna	1969-12-05	17000	student_service
23	8	Keegan	Izzy	1980-06-13	15000	student_service
24	9	Sylvanus	Mathilda	1986-01-10	14000	student_service
25	10	Marla	Sarah	1973-09-25	18000	student_service
26	11	Kitty	Merletta	1965-07-04	19000	student_service
27	12	Ellery	Ayden	1995-08-31	13000	student_service
28	13	Woodie	Darla	1980-02-24	16000	student_service
29	14	Karter	Jennica	1981-10-01	17000	student_service
30	15	Darian	Darcey	1992-11-19	14500	student_service

8. Tabela **ADMINISTRATION _ADDRESS** zawiera adresy korespondencyjne pracowników administracyjnych placówki.



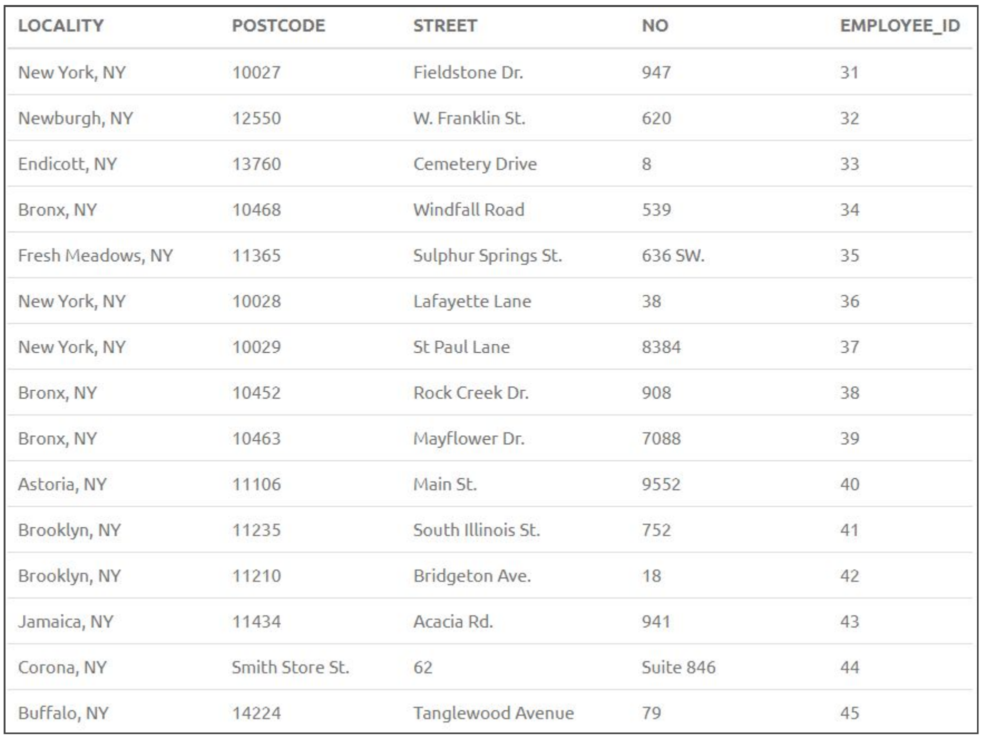
LOCALITY	POSTCODE	STREET	NO	EMPLOYEE_ID
New York, NY	10032	Valley View Street	9486	16
Bronx, NY	10453	Shirley St.	9	17
Jackson Heights, NY	11372	Academy St.	28	18
Brooklyn, NY	11233	East Gates Dr.	73	19
Westbury, NY	11590	N. Augusta Street	85	20
New York, NY	10016	Cambridge Road	922	21
Brentwood, NY	11717	Locust St.	811	22
East Elmhurst, NY	11370	Manhattan Street	108	23
Fresh Meadwos, NY	11365	Ann Lane	7239	24
Brooklyn, NY	11209	Glendale Ave.	39	25
Lockport, NY	14094	Applegate St.	39	26
Fresh Meadows, NY	11365	Cactus Ave.	7427	27
Jamestown, NY	14701	Bayport Street	704 NW.	28
Jamaica, NY	11434	Garfield Rd.	884	29
Buffalo, NY	14215	Foster Court	9700	30

9. Tabela **MAINTENANCE** zawiera informacje o pracownikach porządkowych placówki.



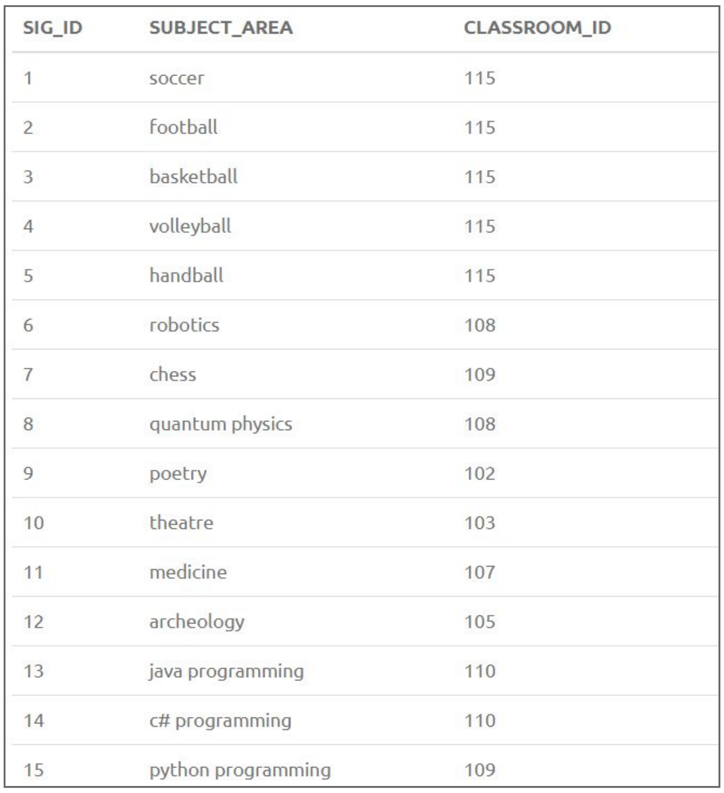
EMPLOYEE_ID	MAINTENANCE_ID	SURNAME	NAME	DATE_OF_BIRTH	SALARY	POSITION
31	1	Kassidy	Brian	1974-03-17	10000	janitor
32	2	Beulah	Robert	1967-11-08	9000	janitor
33	3	Wilbur	Brandon	1971-07-22	10000	janitor
34	4	Carly	Thomas	1979-05-29	9000	janitor
35	5	Shayla	Kevin	1966-08-02	9500	janitor
36	6	Briony	Kathy	1976-08-14	10000	cleaner
37	7	Victor	Bella	1980-12-27	9000	cleaner
38	8	Clarity	Margareth	1978-01-06	9500	cleaner
39	9	Kade	Clarice	1979-10-24	9000	cleaner
40	10	Fox	Jillie	1986-06-26	9500	cleaner
41	11	Asher	Luvinia	1983-11-04	8000	cleaner
42	12	Cullen	Dena	1987-01-18	9000	cleaner
43	13	Wes	Amelia	1994-08-23	8000	cleaner
44	14	Lynsey	Roger	1968-02-15	12000	repairer
45	15	Baldric	Garret	1977-09-13	12000	repairer

10. Tabela **MAINTENANCE _ADDRESS** zawiera adresy korespondencyjne pracowników porządkowych placówki.



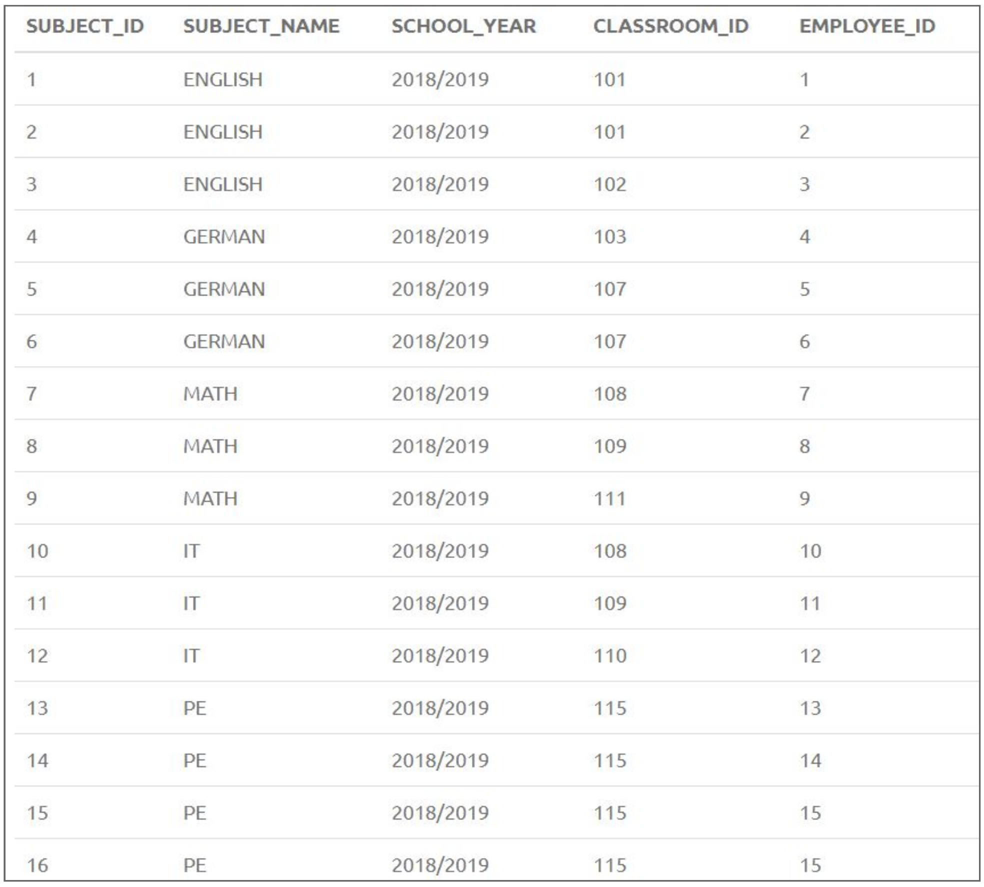
LOCALITY	POSTCODE	STREET	NO	EMPLOYEE_ID
New York, NY	10027	Fieldstone Dr.	947	31
Newburgh, NY	12550	W. Franklin St.	620	32
Endicott, NY	13760	Cemetery Drive	8	33
Bronx, NY	10468	Windfall Road	539	34
Fresh Meadows, NY	11365	Sulphur Springs St.	636 SW.	35
New York, NY	10028	Lafayette Lane	38	36
New York, NY	10029	St Paul Lane	8384	37
Bronx, NY	10452	Rock Creek Dr.	908	38
Bronx, NY	10463	Mayflower Dr.	7088	39
Astoria, NY	11106	Main St.	9552	40
Brooklyn, NY	11235	South Illinois St.	752	41
Brooklyn, NY	11210	Bridgeton Ave.	18	42
Jamaica, NY	11434	Acacia Rd.	941	43
Corona, NY	Smith Store St.	62	Suite 846	44
Buffalo, NY	14224	Tanglewood Avenue	79	45

11. Tabela **SPECIAL_INTERESTS_GROUP** zawiera informacje dotyczące kół zainteresowań.



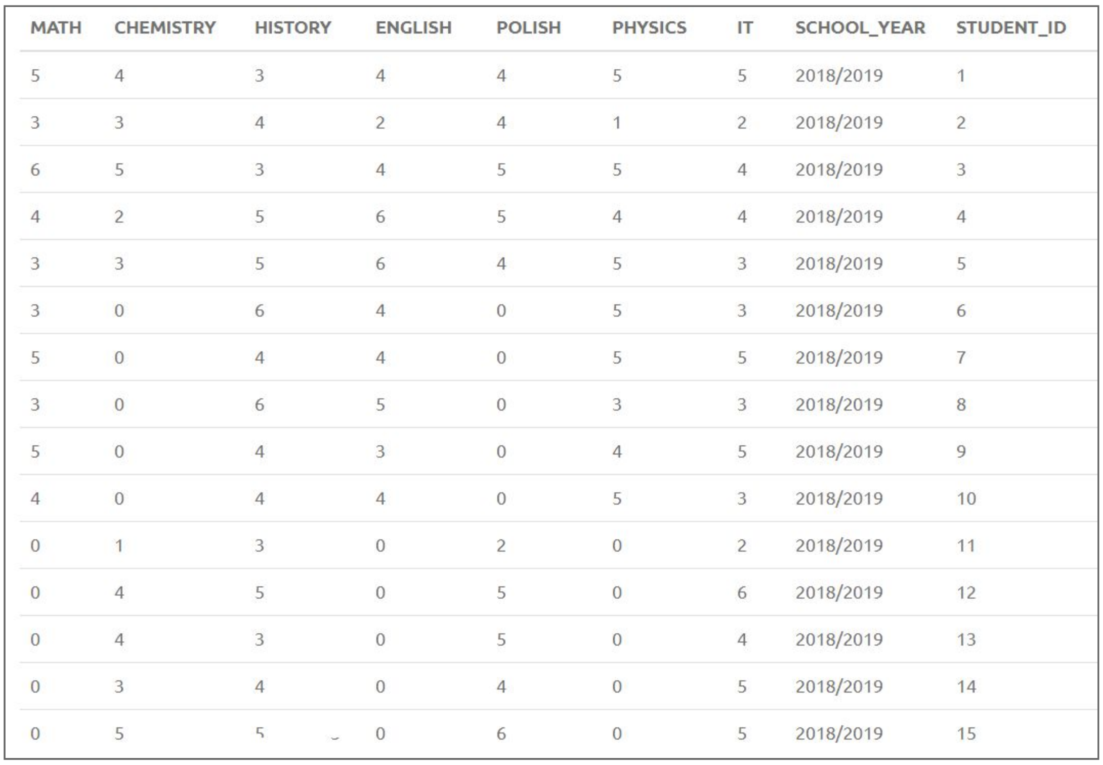
SIG_ID	SUBJECT_AREA	CLASSROOM_ID
1	soccer	115
2	football	115
3	basketball	115
4	volleyball	115
5	handball	115
6	robotics	108
7	chess	109
8	quantum physics	108
9	poetry	102
10	theatre	103
11	medicine	107
12	archeology	105
13	java programming	110
14	c# programming	110
15	python programming	109

12. Tabela **SUBJECT** zawiera informacje dotyczące przedmiotów szkolnych.



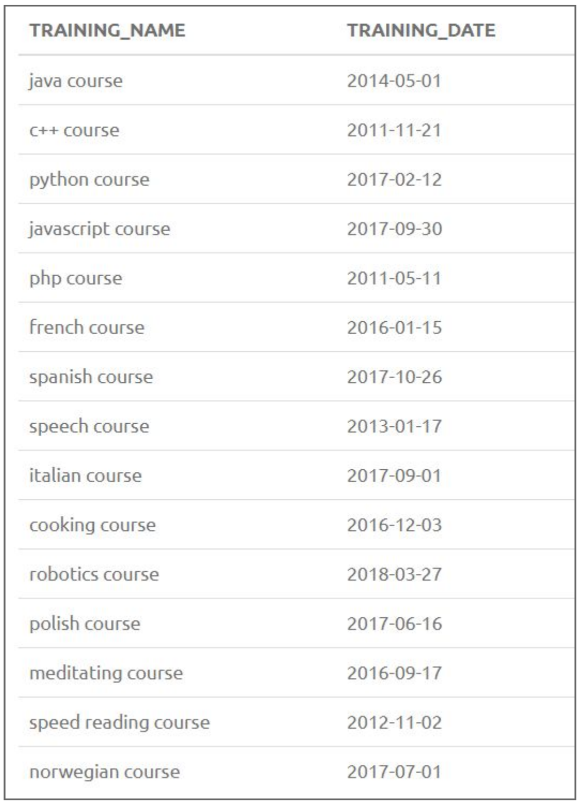
SUBJECT_ID	SUBJECT_NAME	SCHOOL_YEAR	CLASSROOM_ID	EMPLOYEE_ID
1	ENGLISH	2018/2019	101	1
2	ENGLISH	2018/2019	101	2
3	ENGLISH	2018/2019	102	3
4	GERMAN	2018/2019	103	4
5	GERMAN	2018/2019	107	5
6	GERMAN	2018/2019	107	6
7	MATH	2018/2019	108	7
8	MATH	2018/2019	109	8
9	MATH	2018/2019	111	9
10	IT	2018/2019	108	10
11	IT	2018/2019	109	11
12	IT	2018/2019	110	12
13	PE	2018/2019	115	13
14	PE	2018/2019	115	14
15	PE	2018/2019	115	15
16	PE	2018/2019	115	15

13. Tabela **FINAL_ASSESSMENT** zawiera informacje dotyczące ocen uczniów placówki.



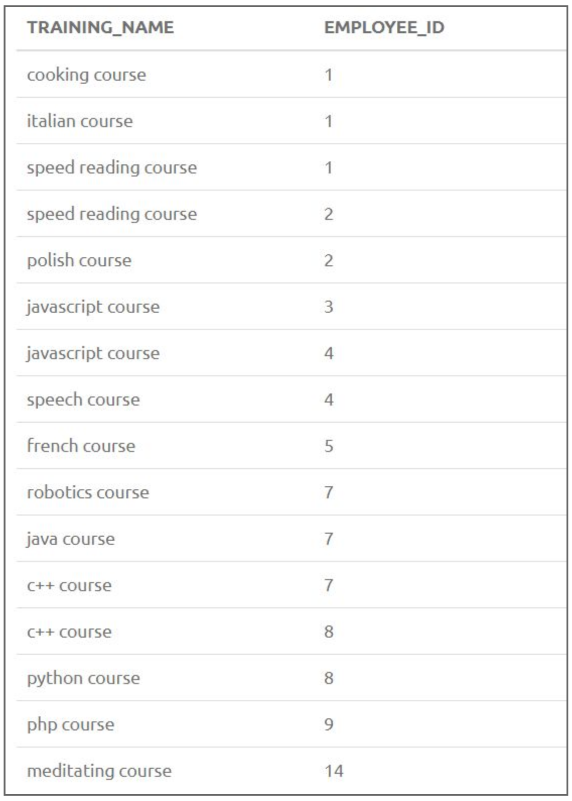
MATH	CHEMISTRY	HISTORY	ENGLISH	POLISH	PHYSICS	IT	SCHOOL_YEAR	STUDENT_ID
5	4	3	4	4	5	5	2018/2019	1
3	3	4	2	4	1	2	2018/2019	2
6	5	3	4	5	5	4	2018/2019	3
4	2	5	6	5	4	4	2018/2019	4
3	3	5	6	4	5	3	2018/2019	5
3	0	6	4	0	5	3	2018/2019	6
5	0	4	4	0	5	5	2018/2019	7
3	0	6	5	0	3	3	2018/2019	8
5	0	4	3	0	4	5	2018/2019	9
4	0	4	4	0	5	3	2018/2019	10
0	1	3	0	2	0	2	2018/2019	11
0	4	5	0	5	0	6	2018/2019	12
0	4	3	0	5	0	4	2018/2019	13
0	3	4	0	4	0	5	2018/2019	14
0	5	5	0	6	0	5	2018/2019	15

14. Tabela **TRAINING_COMPLETED** zawiera informacje o zamkniętych dodatkowych kursach.



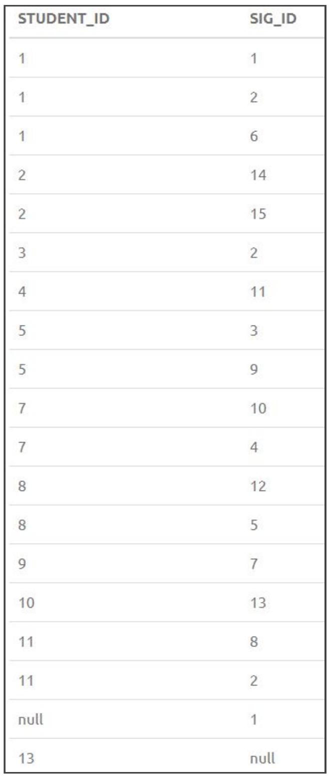
TRAINING_NAME	TRAINING_DATE
java course	2014-05-01
c++ course	2011-11-21
python course	2017-02-12
javascript course	2017-09-30
php course	2011-05-11
french course	2016-01-15
spanish course	2017-10-26
speech course	2013-01-17
italian course	2017-09-01
cooking course	2016-12-03
robotics course	2018-03-27
polish course	2017-06-16
meditating course	2016-09-17
speed reading course	2012-11-02
norwegian course	2017-07-01

15. Tabela **TEACHER_TRAINING** zawiera informacje o pracownikach, którzy odbyli kursy.



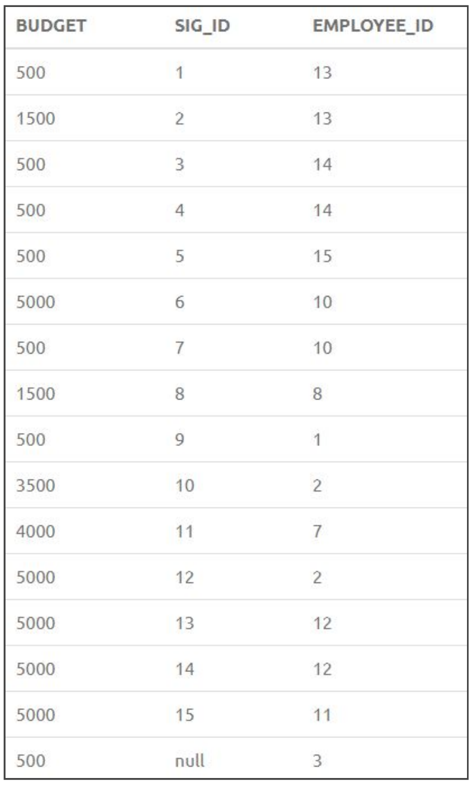
TRAINING_NAME	EMPLOYEE_ID
cooking course	1
italian course	1
speed reading course	1
speed reading course	2
polish course	2
javascript course	3
javascript course	4
speech course	4
french course	5
robotics course	7
java course	7
c++ course	7
c++ course	8
python course	8
php course	9
meditating course	14

16. Tabela **STUDENT_SIG** łączy uczniów z kołami zainteresowań, w których uczestniczą.



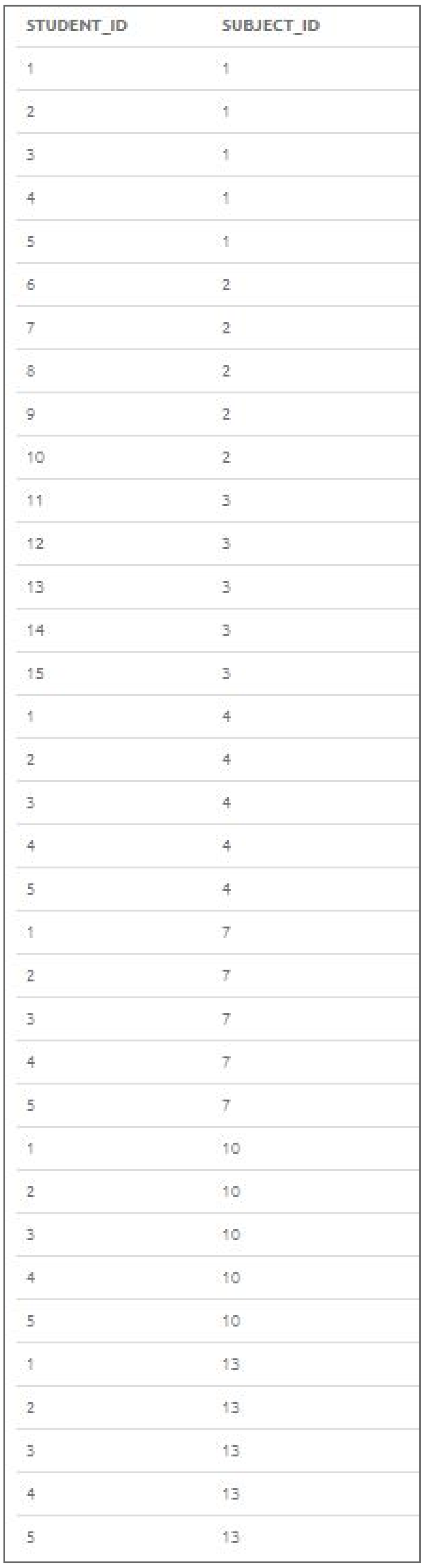
STUDENT_ID	SIG_ID
1	1
1	2
1	6
2	14
2	15
3	2
4	11
5	3
5	9
7	10
7	4
8	12
8	5
9	7
10	13
11	8
11	2
null	1
13	null

17. Tabela **TEACHER_SIG** łączy nauczycieli z kołami zainteresowań, które prowadzą.



BUDGET	SIG_ID	EMPLOYEE_ID
500	1	13
1500	2	13
500	3	14
500	4	14
500	5	15
5000	6	10
500	7	10
1500	8	8
500	9	1
3500	10	2
4000	11	7
5000	12	2
5000	13	12
5000	14	12
5000	15	11
500	null	3

18. Tabela **STUDENT_SUBJECT** łączy uczniów z przedmiotami, w których uczestniczą.

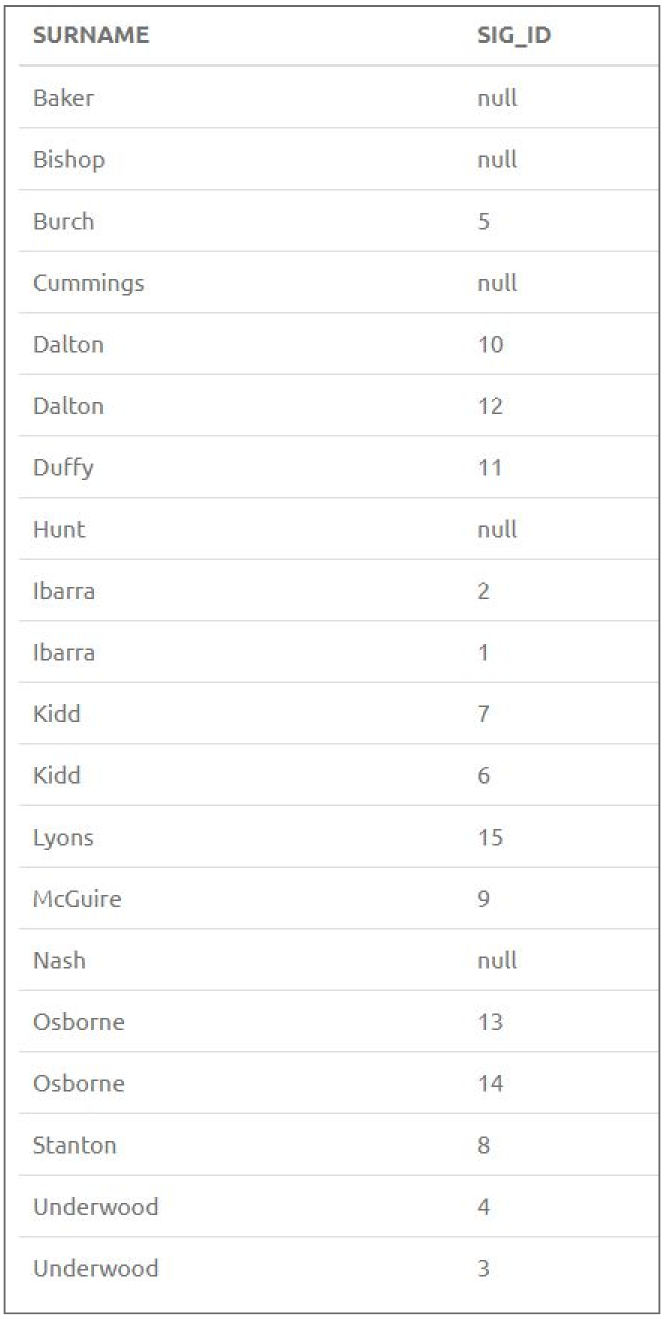


STUDENT_ID	SUBJECT_ID
1	1
2	1
3	1
4	1
5	1
6	2
7	2
8	2
9	2
10	2
11	3
12	3
13	3
14	3
15	3
1	4
2	4
3	4
4	4
5	4
1	7
2	7
3	7
4	7
5	7
1	10
2	10
3	10
4	10
5	10
1	13
2	13
3	13
4	13
5	13

ZAPYTANIA SQL, ODPOWIEDZI NA ZAPYTANIA

1. Zapytanie wyświetli nauczycieli z ich zajęciami dodatkowymi.
Jeśli nauczyciel nie prowadzi żadnych zajęć dodatwkoych to zostanie zwrócona wartość null.

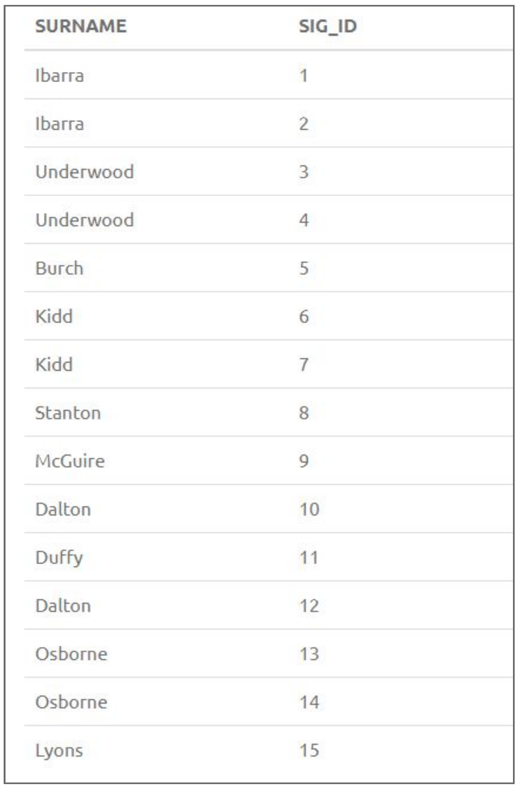
```
select "TEACHER"."SURNAME", "TEACHER_SIG"."SIG_ID"
from"TEACHER"
full outer join "TEACHER_SIG"
on "TEACHER"."EMPLOYEE_ID"="TEACHER_SIG"."EMPLOYEE_ID"
order by "TEACHER"."SURNAME"
```



SURNAME	SIG_ID
Baker	null
Bishop	null
Burch	5
Cummings	null
Dalton	10
Dalton	12
Duffy	11
Hunt	null
Ibarra	2
Ibarra	1
Kidd	7
Kidd	6
Lyons	15
McGuire	9
Nash	null
Osborne	13
Osborne	14
Stanton	8
Underwood	4
Underwood	3

2. Zapytanie wyświetli wyłącznie nauczycieli, którzy prowadzą co najmniej jedne zajęcia dodatkowe dla uczniów.

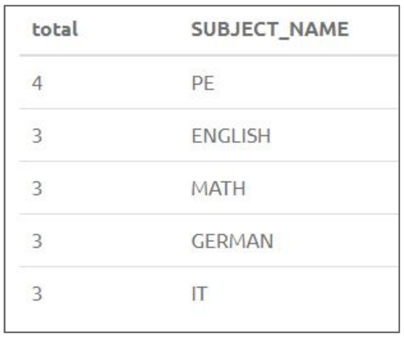
```
select "TEACHER"."SURNAME", "TEACHER_SIG"."SIG_ID"
from"TEACHER"
inner join "TEACHER_SIG"
on "TEACHER"."EMPLOYEE_ID"="TEACHER_SIG"."EMPLOYEE_ID"
```



SURNAME	SIG_ID
Ibarra	1
Ibarra	2
Underwood	3
Underwood	4
Burch	5
Kidd	6
Kidd	7
Stanton	8
McGuire	9
Dalton	10
Duffy	11
Dalton	12
Osborne	13
Osborne	14
Lyons	15

3. Zapytanie wyznacza, ilu nauczycieli jest przypisanych do danych przedmiotów, a następnie sortuje powyższe dane malejąco.

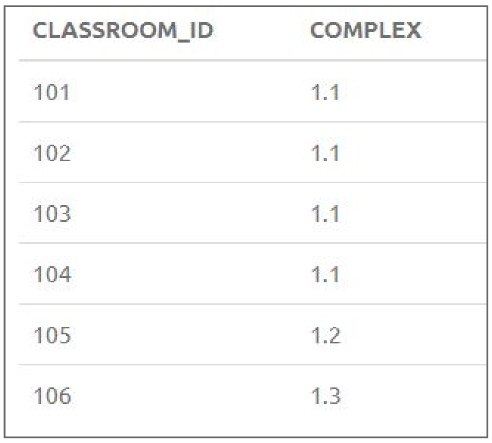
```
select count(“EMPLOYEE_ID”) as total,  “SUBJECT_NAME” from “SUBJECT”
group by “SUBJECT_NAME” having count(“EMPLOYEE_ID”) >= 3
order by count(“EMPLOYEE_ID”) desc
```



total	SUBJECT_NAME
4	PE
3	ENGLISH
3	MATH
3	GERMAN
3	IT

4. Zapytanie wyznacza wszystkie sale znajdujące się w budynkach 1.1, 1.2, 1.3.

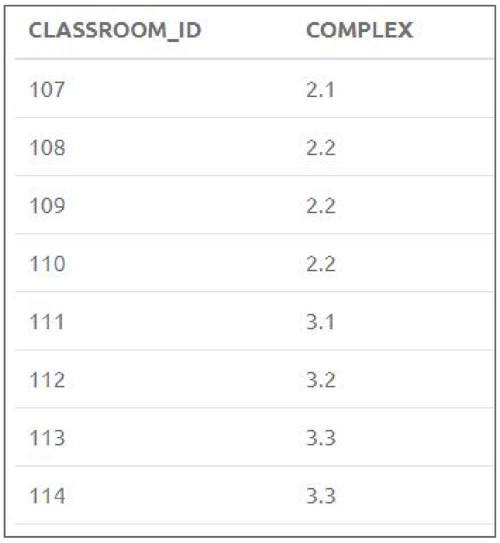
```
select * from “CLASSROOM” where “COMPLEX” in (‘1.1’, ‘1.2’, ‘1.3’)
```



CLASSROOM_ID	COMPLEX
101	1.1
102	1.1
103	1.1
104	1.1
105	1.2
106	1.3

5. Zapytanie wyznacza wszystkie sale, które nie znajdują się w budynkach 1.1, 1.2, 1.3.

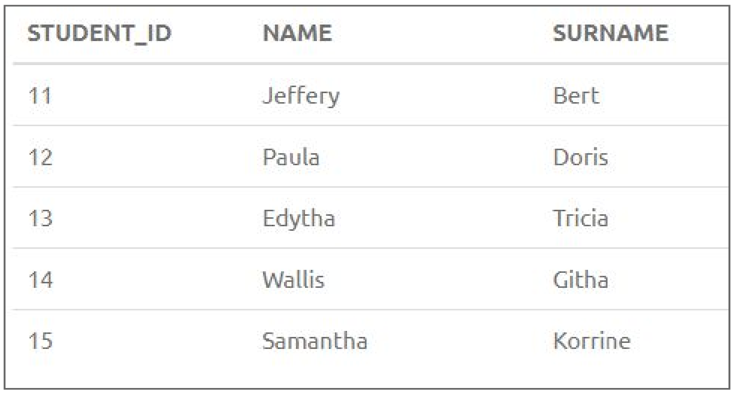
```
select * from “CLASSROOM” where “COMPLEX” not in (‘1.1’, ‘1.2’, ‘1.3’)
```



CLASSROOM_ID	COMPLEX
107	2.1
108	2.2
109	2.2
110	2.2
111	3.1
112	3.2
113	3.3
114	3.3

6. Zapytanie sprawdza, czy są uczniowie z oceną niedostateczną z matematyki.

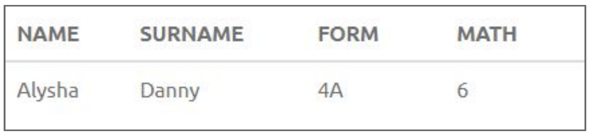
```
select “STUDENT”.”STUDENT_ID”, “STUDENT”.”NAME”“STUDENT”.”SURNAME”
from “STUDENT” where “STUDENT”.”STUDENT_ID” = any (select “STUDENT_ID” from
“FINAL_ASSESSMENT” where “MATH” < 2)
```



STUDENT_ID	NAME	SURNAME
11	Jeffery	Bert
12	Paula	Doris
13	Edytha	Tricia
14	Wallis	Githa
15	Samantha	Korrine

7. Zapytanie wyszukuje uczniów z najwyższymi ocenami z matematyki.

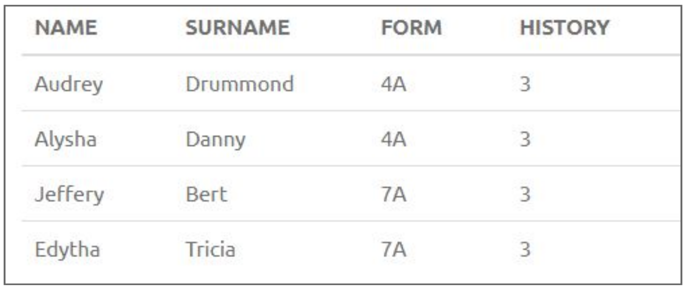
```
select "STUDENT"."NAME", "STUDENT"."SURNAME", "STUDENT"."FORM",
"FINAL_ASSESSMENT"."MATH" from "STUDENT" full join "FINAL_ASSESSMENT"
on "STUDENT"."STUDENT_ID" = "FINAL_ASSESSMENT"."STUDENT_ID"
where "MATH" >= all (select "MATH" from "FINAL_ASSESSMENT")
```



NAME	SURNAME	FORM	MATH
Alysha	Danny	4A	6

8. Zapytanie wyszukuje uczniów z najniższymi ocenami z historii.

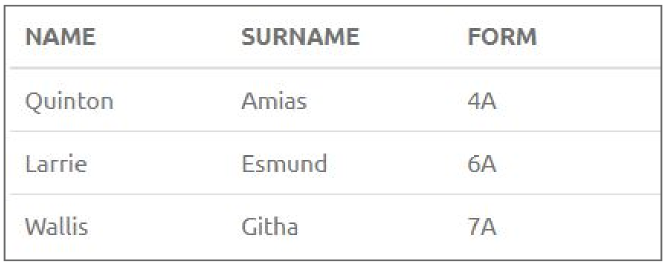
```
select "STUDENT"."NAME", "STUDENT"."SURNAME", "STUDENT"."FORM",
"FINAL_ASSESSMENT"."HISTORY" from "STUDENT" full join "FINAL_ASSESSMENT"
on "STUDENT"."STUDENT_ID" = "FINAL_ASSESSMENT"."STUDENT_ID"
where "HISTORY" <= all (select "HISTORY" from "FINAL_ASSESSMENT")
```



NAME	SURNAME	FORM	HISTORY
Audrey	Drummond	4A	3
Alysha	Danny	4A	3
Jeffery	Bert	7A	3
Edytha	Tricia	7A	3

9. Zapytanie wyswietla nazwiska uczniów, którzy otrzymają świadectwo z paskiem.

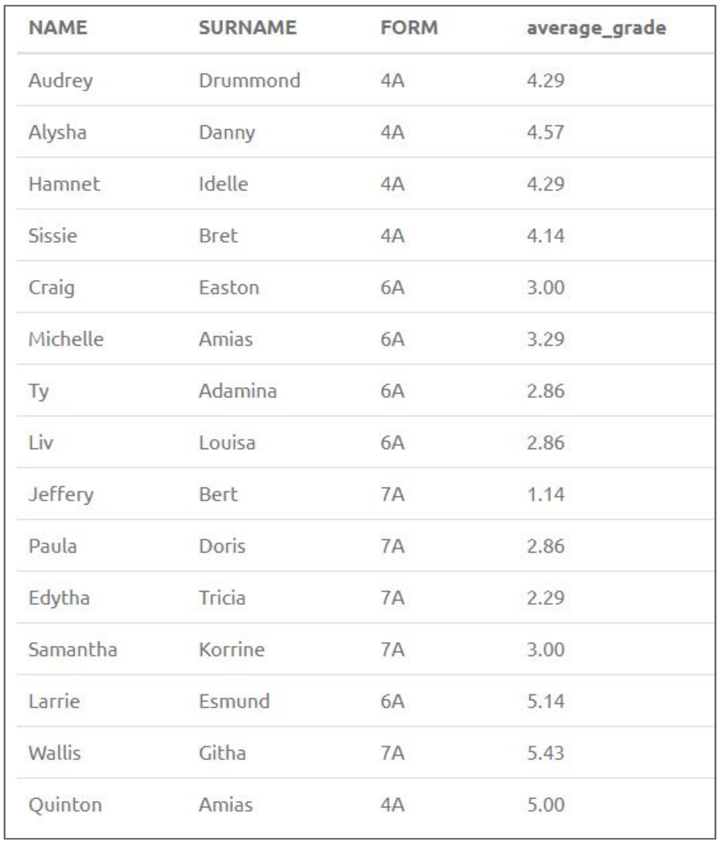
```
select "STUDENT"."NAME", "STUDENT"."SURNAME", "STUDENT"."FORM" from "STUDENT"
where exists (select * from "FINAL_ASSESSMENT"
where "STUDENT"."STUDENT_ID" = "FINAL_ASSESSMENT"."STUDENT_ID" and
(("MATH" + "CHEMISTRY" + "HISTORY" + "ENGLISH" + "POLISH" + "PHYSICS" + "IT")/7) > 4.75)
```



NAME	SURNAME	FORM
Quinton	Amias	4A
Larrie	Esmund	6A
Wallis	Githa	7A

10. Zapytanie podstawowe dane o uczniach oraz średnie ocen jakie uzyskali.

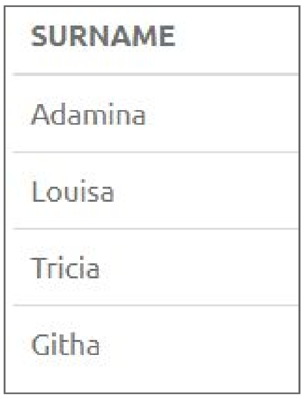
```
select "STUDENT"."NAME", "STUDENT"."SURNAME", "STUDENT"."FORM", to_char((("MATH" +
"CHEMISTRY" + "HISTORY" + "ENGLISH" + "POLISH" + "PHYSICS" + "IT")/7.0), '9D99') as
average_grade from "STUDENT" inner join "FINAL_ASSESSMENT" on
"STUDENT"."STUDENT_ID" = "FINAL_ASSESSMENT"."STUDENT_ID"
```



NAME	SURNAME	FORM	average_grade
Audrey	Drummond	4A	4.29
Alysha	Danny	4A	4.57
Hamnet	Idelle	4A	4.29
Sissie	Bret	4A	4.14
Craig	Easton	6A	3.00
Michelle	Amias	6A	3.29
Ty	Adamina	6A	2.86
Liv	Louisa	6A	2.86
Jeffery	Bert	7A	1.14
Paula	Doris	7A	2.86
Edytha	Tricia	7A	2.29
Samantha	Korrine	7A	3.00
Larrie	Esmund	6A	5.14
Wallis	Githa	7A	5.43
Quinton	Amias	4A	5.00

11. Zapytanie zwraca nazwiska tych uczniów, których nazwiska kończą się na literę ‘a’.

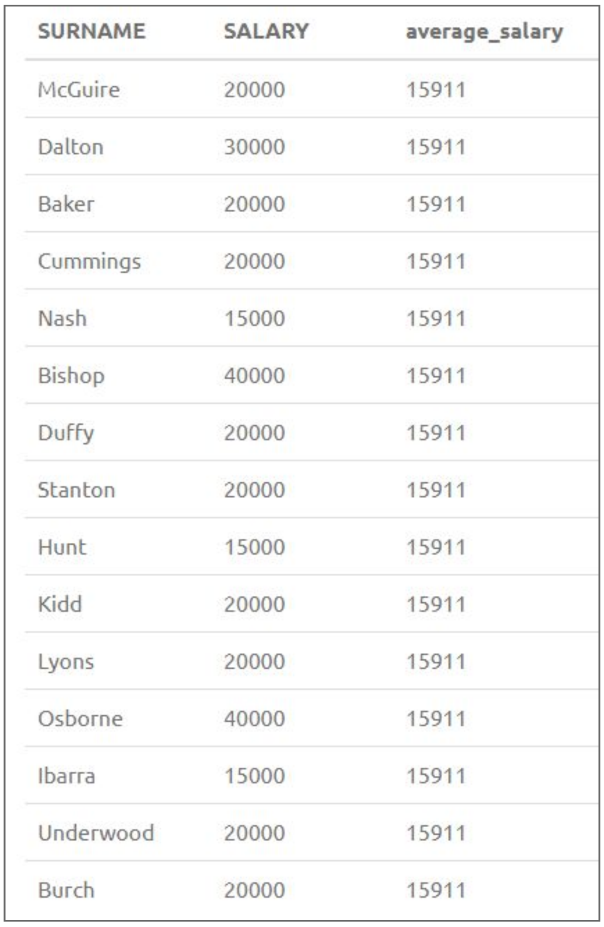
select "SURNAME" **from** "STUDENT" **where** "SURNAME" **like '%a'**



SURNAME
Adamina
Louisa
Tricia
Githa

12. Zapytanie zwraca nazwiska nauczycieli, ich zarobki oraz średnią zarobków wszystkich pracowników szkoły.

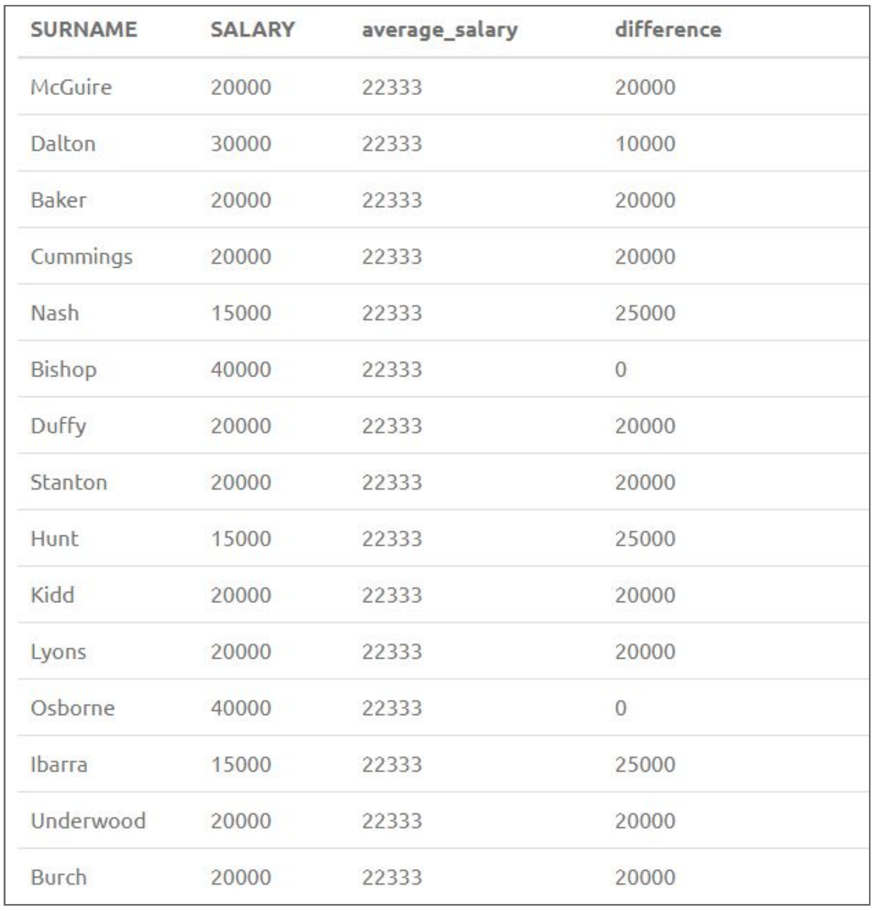
select "SURNAME", "SALARY", **to_char**((**select avg** (("TEACHER"."SALARY" + "ADMINISTRATION"."SALARY" + "MAINTENANCE"."SALARY")/3) **from**"TEACHER", "ADMINISTRATION", "MAINTENANCE"), '99999') **as** average_salary **from**"TEACHER"



SURNAME	SALARY	average_salary
McGuire	20000	15911
Dalton	30000	15911
Baker	20000	15911
Cummings	20000	15911
Nash	15000	15911
Bishop	40000	15911
Duffy	20000	15911
Stanton	20000	15911
Hunt	15000	15911
Kidd	20000	15911
Lyons	20000	15911
Osborne	40000	15911
Ibarra	15000	15911
Underwood	20000	15911
Burch	20000	15911

13. Zapytanie zwraca nazwiska nauczycieli, ich zarobki, średnią zarobków nauczycieli oraz różnicę między zarobkami danego nauczyciela a maksymalnymi zarobkami z działu nauczycielskiego.

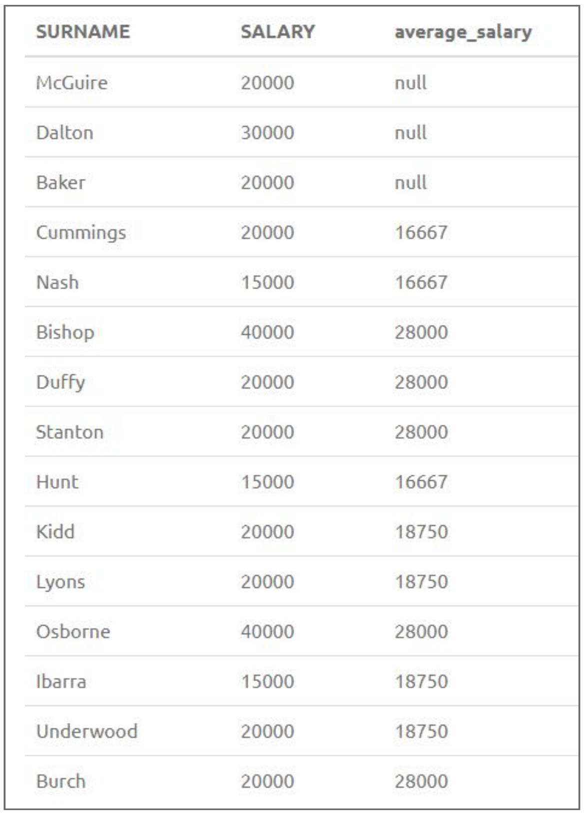
select "SURNAME", "SALARY", **to_char**((**select avg** ("SALARY") **from** "TEACHER"), '99999') **as** average_salary, ((**select max**("SALARY") **from** "TEACHER") - "SALARY") **as** difference **from** "TEACHER"



SURNAME	SALARY	average_salary	difference
McGuire	20000	22333	20000
Dalton	30000	22333	10000
Baker	20000	22333	20000
Cummings	20000	22333	20000
Nash	15000	22333	25000
Bishop	40000	22333	0
Duffy	20000	22333	20000
Stanton	20000	22333	20000
Hunt	15000	22333	25000
Kidd	20000	22333	20000
Lyons	20000	22333	20000
Osborne	40000	22333	0
Ibarra	15000	22333	25000
Underwood	20000	22333	20000
Burch	20000	22333	20000

14. Zapytanie zwraca nazwiska nauczycieli, ich zarobki oraz średnią zarobków wśród nauczycieli, którzy są podwładni temu samemu dyrektorowi, co badany.

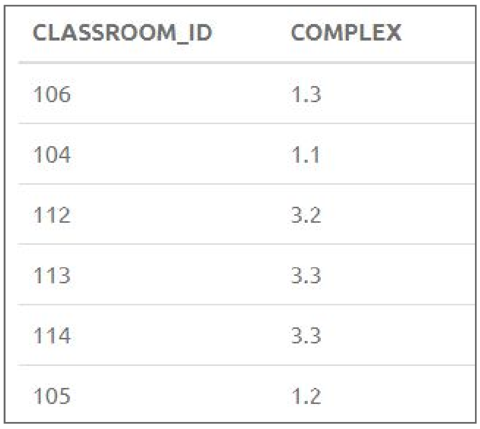
select "SURNAME", "SALARY", **to_char**((**select avg** ("SALARY") **from** "TEACHER" **where** "EMPLOYEE_SUPERVISOR_ID" = T."EMPLOYEE_SUPERVISOR_ID"), '99999') **as** average_salary **from** "TEACHER" T



SURNAME	SALARY	average_salary
McGuire	20000	null
Dalton	30000	null
Baker	20000	null
Cummings	20000	16667
Nash	15000	16667
Bishop	40000	28000
Duffy	20000	28000
Stanton	20000	28000
Hunt	15000	16667
Kidd	20000	18750
Lyons	20000	18750
Osborne	40000	28000
Ibarra	15000	18750
Underwood	20000	18750
Burch	20000	28000

15. Zapytanie zwraca informacje na temat sal, które nie zostały jeszcze wykorzystane ani razu.

select "CLASSROOM"."CLASSROOM_ID", "CLASSROOM"."COMPLEX"
from "CLASSROOM" **full outer join** "SUBJECT" **on** "CLASSROOM"."CLASSROOM_ID" = "SUBJECT"."CLASSROOM_ID" **where not exists** (**select** * **from** "SUBJECT"
where "SUBJECT"."CLASSROOM_ID" = "CLASSROOM"."CLASSROOM_ID")



CLASSROOM_ID	COMPLEX
106	1.3
104	1.1
112	3.2
113	3.3
114	3.3
105	1.2