BUCHAREST UNIVERSITY OF ECONOMIC STUDIES

CYBERNETICS, STATISTICS AND ECONOMIC INFORMATICS FACULTY

DATABASES PROJECT

Managing a Cat Contest

Coordinator:

Conf. Univ. Dr. BELCIU Anda

Student: COSTAN CRISTIANA

Group: 1066

Bucharest

2022

CONTENTS

DESCRIPTION OF THE PROJECT

My project was created having in mind the idea of managing a cat contest. Due to the large number of data about the cats, judges and owners, one can manage more easily with a database the ranking and grouping of the cats based on their performance and characteristics along with the sorting of the judges based on their years of experience or other attributes related to their profession.

The relationships between tables can be noticed in the database schema shown in the following chapter. When it comes to speaking about owners, only one owner can come from each registered address and an address can belong to one owner only, being an one to one relationship between the tables addresses and owners. This type of relationship was chosen as a precautionary measure: we long for diversity and we do not seek to cause hatred between neighbors.

Speaking of relationships, an one to many relationship is the one between the cats table and the owners table: a cat can have an owner, while an owner can have many cats. Another one to many relationship is the recursive relationship from the judges table, based on the manager ID, which also establishes a hierarchy between the judges. The rest of the relationships are one to one: a cat can have only one place in the contest and a place in the contest can be occupied by only one cat. Moreover, a cat can be judged only by one judge and a judge can arbitrate only one cat, this being a solid refference to traditional judging practiced in Europe.

Due to having a small number of cats, the contest will focus mainly on evaluating them together and we will not care about their breeds or sizes when it comes to their final grades. We will only take into account their distinctive characteristics (breed, sex, size, fur type) when we either need to group them separately before the contest to avoid possible accidents or to study the participations rate of one particularity compared to another.

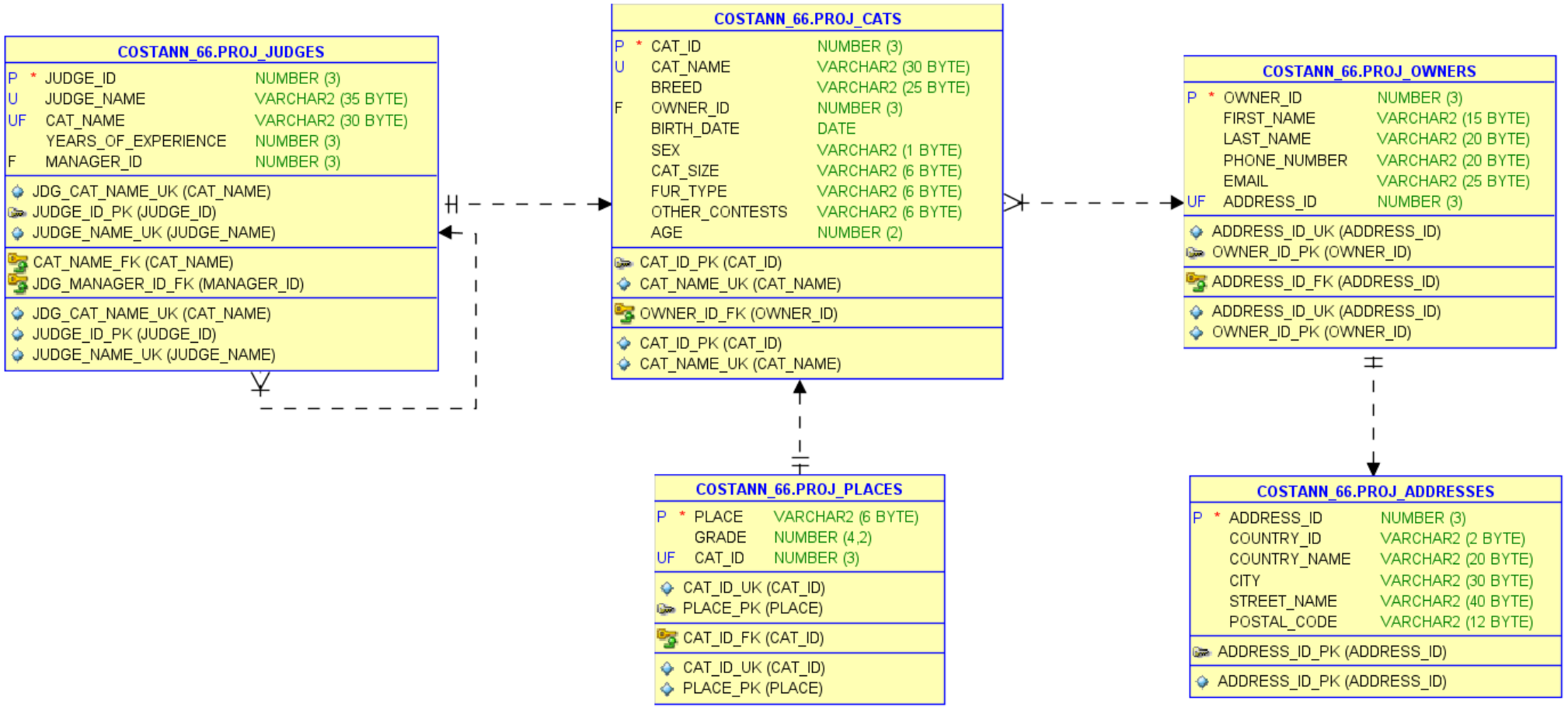
As in every cat contest, the cats have scene names, therefore they need to be unique. Furthermore, the judges that participate in the contest need to judge only one cat, hence their name uniqueness. Because I have taken into consideration the possibility of having one or more owners show up with two or more cats because of their kinship, the owner ID coresponding to the registered cat is not unique.

Additionally, ID related data from the tables judges (e.g. judge name), cats (e.g. cat name) and addresses (e.g. country ID) can be classified independently from other tables by data from the same table, while the others require data from other tables for their grouping to make sense.

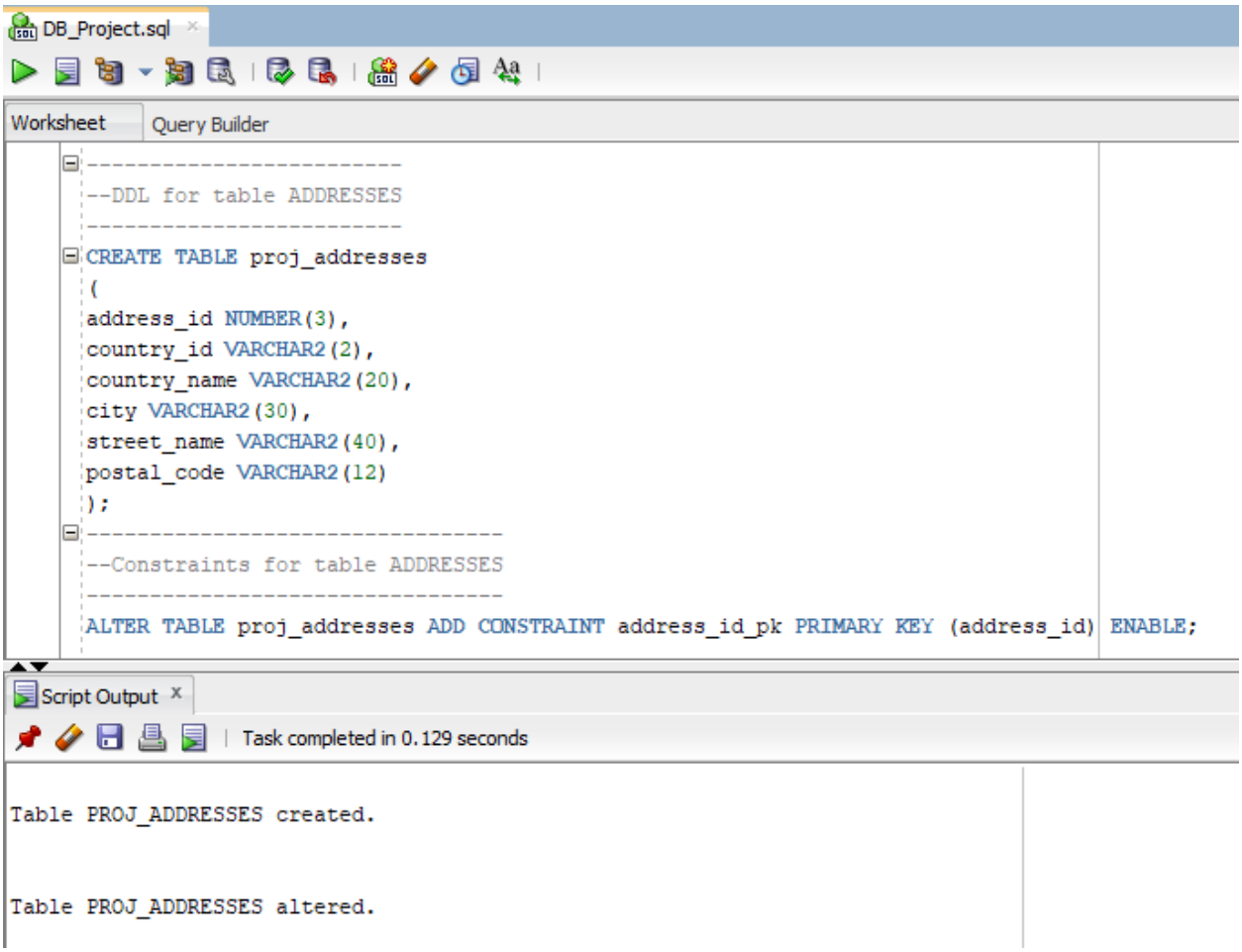
Of course, we cannot forget the awards that a contest is well-known for. That being said, this competition has many title awards and trophies for the cats, given the fact that participants do not join for money.

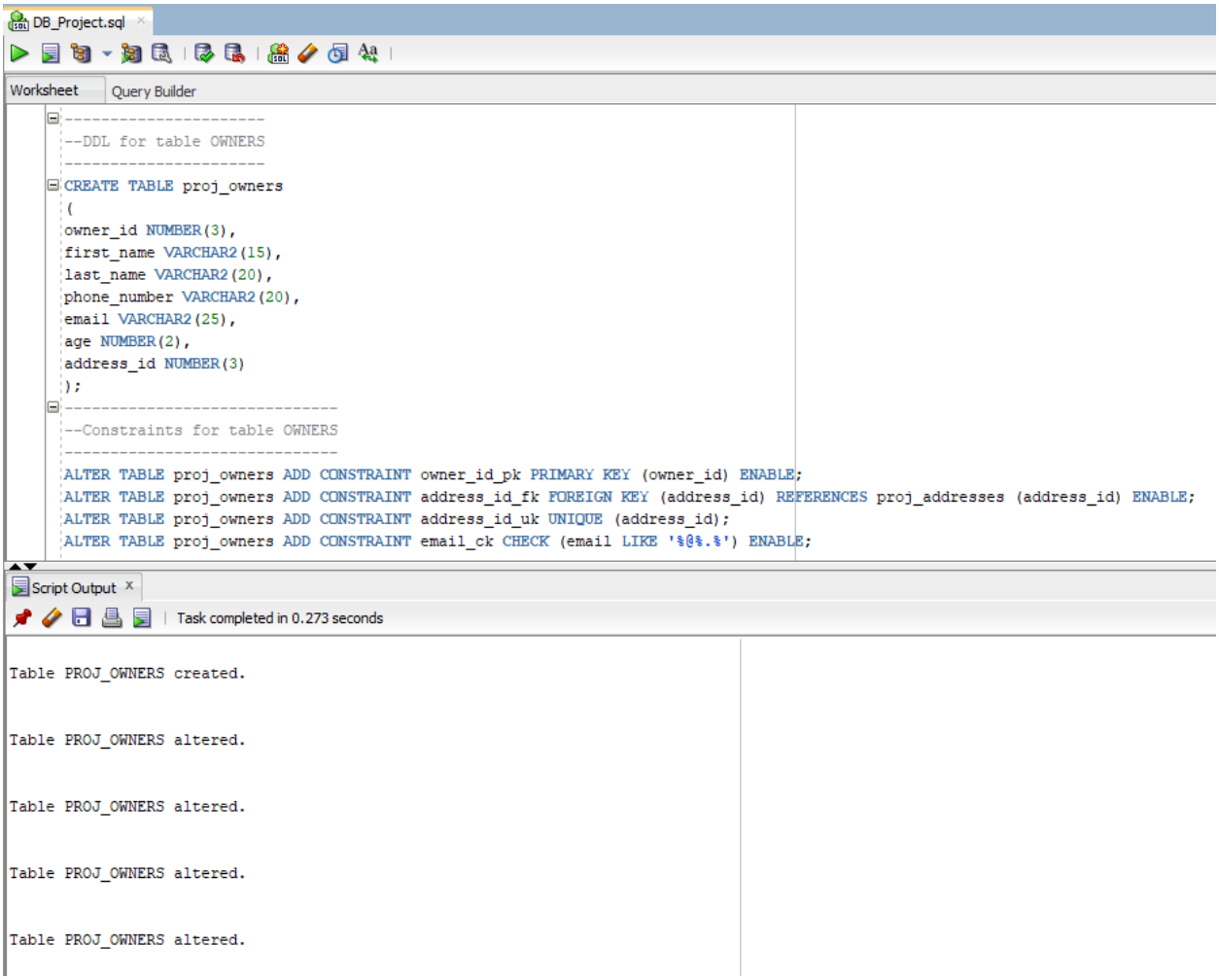
Although some required statements fit better with the use of different numbers, I have tried my best to use them on my data that is mainly composed of cats' specific features or owners' and judges' information.

THE DATABASE SCHEMA



CONSTRUCTING THE DATABASE



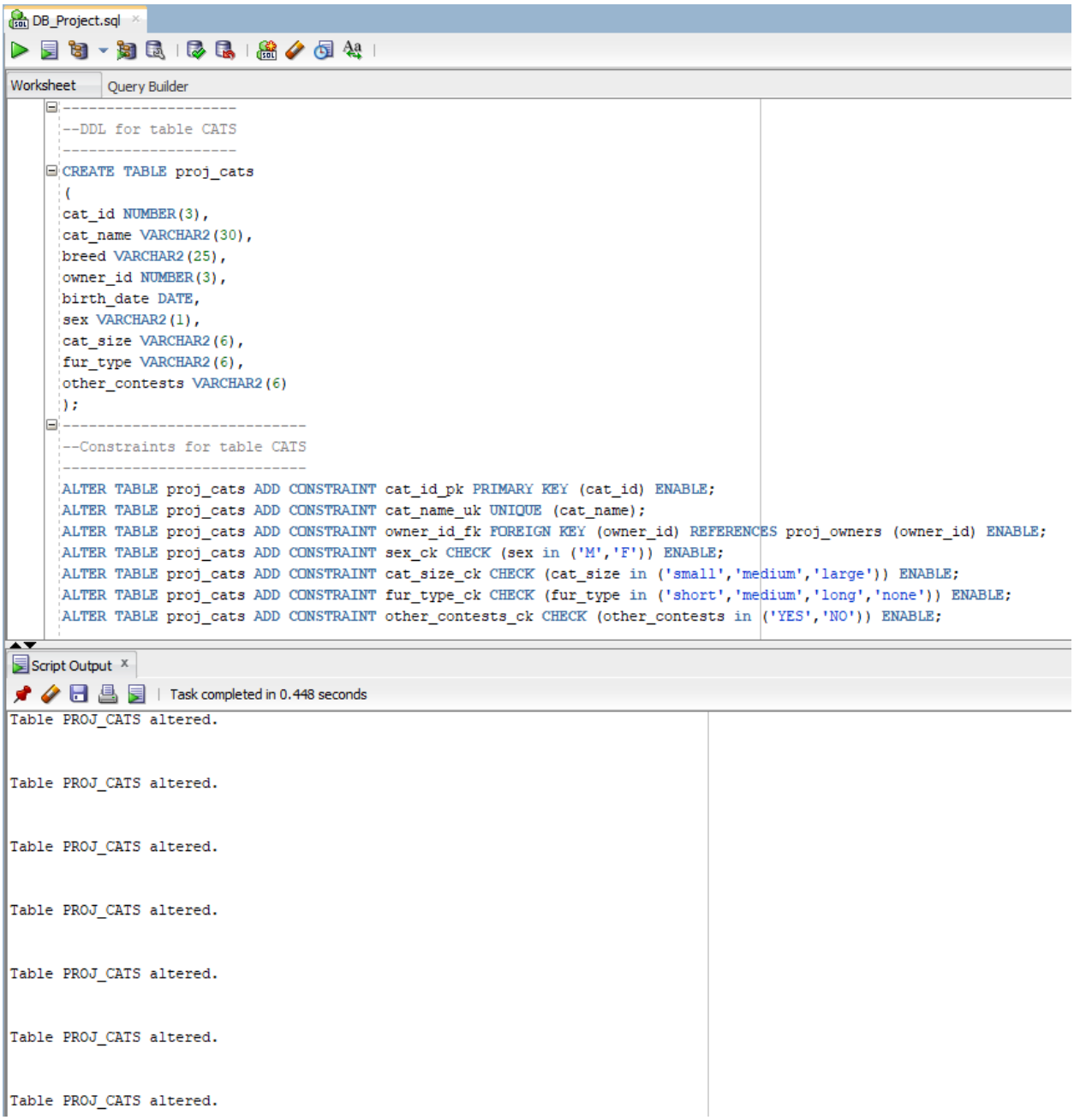

DB_Project.sql
Worksheet
Query Builder

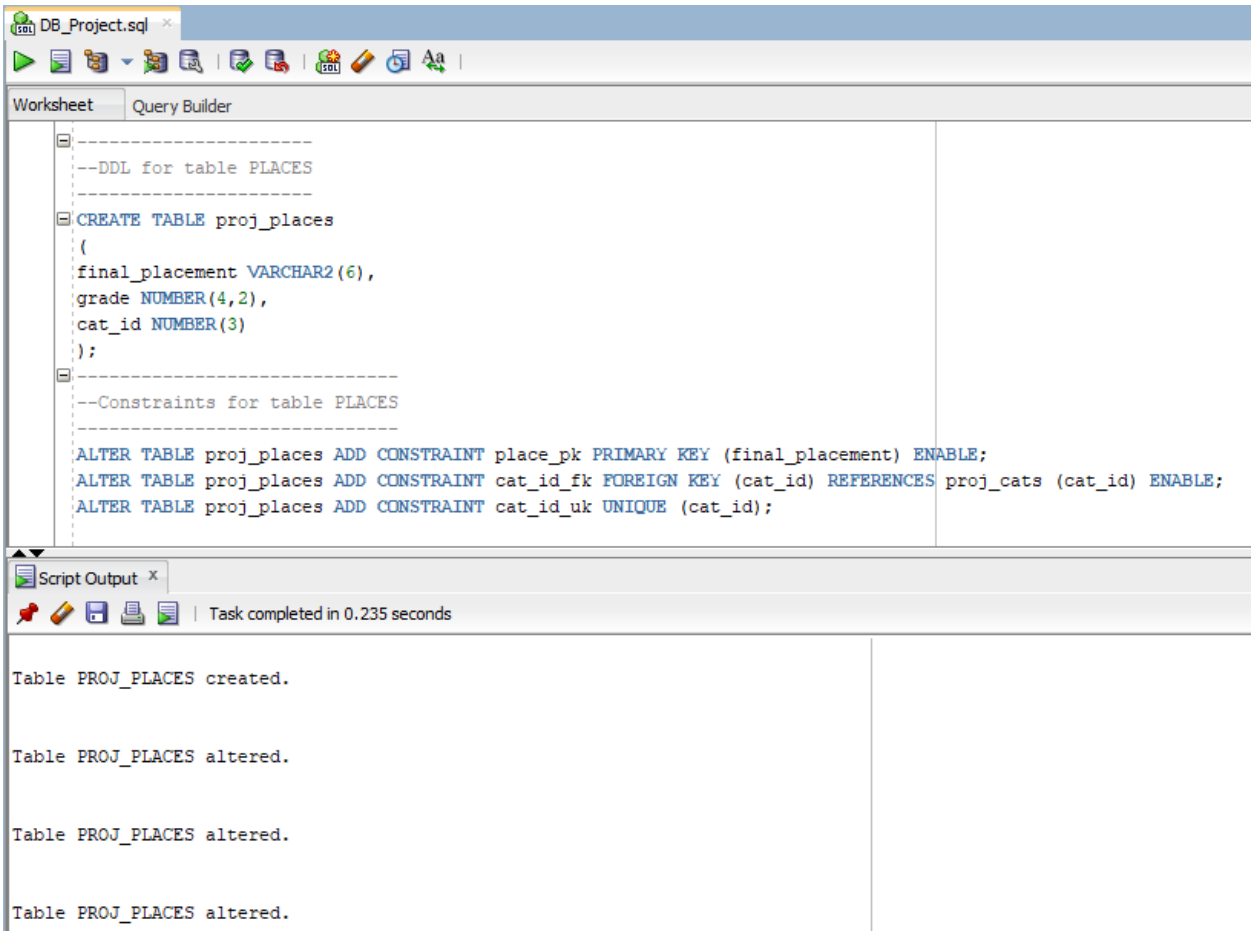
--DDL for table OWNERS

CREATE TABLE proj_owners
(
owner_id NUMBER(3),
first_name VARCHAR2(15),
last_name VARCHAR2(20),
phone_number VARCHAR2(20),
email VARCHAR2(25),
age NUMBER(2),
address_id NUMBER(3)
);

--Constraints for table OWNERS

ALTER TABLE proj_owners ADD CONSTRAINT owner_id_pk PRIMARY KEY (owner_id) ENABLE;
ALTER TABLE proj_owners ADD CONSTRAINT address_id_fk FOREIGN KEY (address_id) REFERENCES proj_addresses (address_id) ENABLE;
ALTER TABLE proj_owners ADD CONSTRAINT address_id_uk UNIQUE (address_id);
ALTER TABLE proj_owners ADD CONSTRAINT email_ck CHECK (email LIKE '%@%.%') ENABLE;
Script Output
Task completed in 0.273 seconds
Table PROJ_OWNERS created.
Table PROJ_OWNERS altered.
Table PROJ_OWNERS altered.
Table PROJ_OWNERS altered.
Table PROJ_OWNERS altered.


DB_Project.sql
Worksheet
Query Builder
--DDL for table CATS
CREATE TABLE proj_cats
(
cat_id NUMBER(3),
cat_name VARCHAR2(30),
breed VARCHAR2(25),
owner_id NUMBER(3),
birth_date DATE,
sex VARCHAR2(1),
cat_size VARCHAR2(6),
fur_type VARCHAR2(6),
other_contests VARCHAR2(6)
);
--Constraints for table CATS
ALTER TABLE proj_cats ADD CONSTRAINT cat_id_pk PRIMARY KEY (cat_id) ENABLE;
ALTER TABLE proj_cats ADD CONSTRAINT cat_name_uk UNIQUE (cat_name);
ALTER TABLE proj_cats ADD CONSTRAINT owner_id_fk FOREIGN KEY (owner_id) REFERENCES proj_owners (owner_id) ENABLE;
ALTER TABLE proj_cats ADD CONSTRAINT sex_ck CHECK (sex in ('M','F')) ENABLE;
ALTER TABLE proj_cats ADD CONSTRAINT cat_size_ck CHECK (cat_size in ('small','medium','large')) ENABLE;
ALTER TABLE proj_cats ADD CONSTRAINT fur_type_ck CHECK (fur_type in ('short','medium','long','none')) ENABLE;
ALTER TABLE proj_cats ADD CONSTRAINT other_contests_ck CHECK (other_contests in ('YES','NO')) ENABLE;
Script Output
Task completed in 0.448 seconds
Table PROJ_CATS altered.
Table PROJ_CATS altered.
Table PROJ_CATS altered.
Table PROJ_CATS altered.
Table PROJ_CATS altered.
Table PROJ_CATS altered.
Table PROJ_CATS altered.

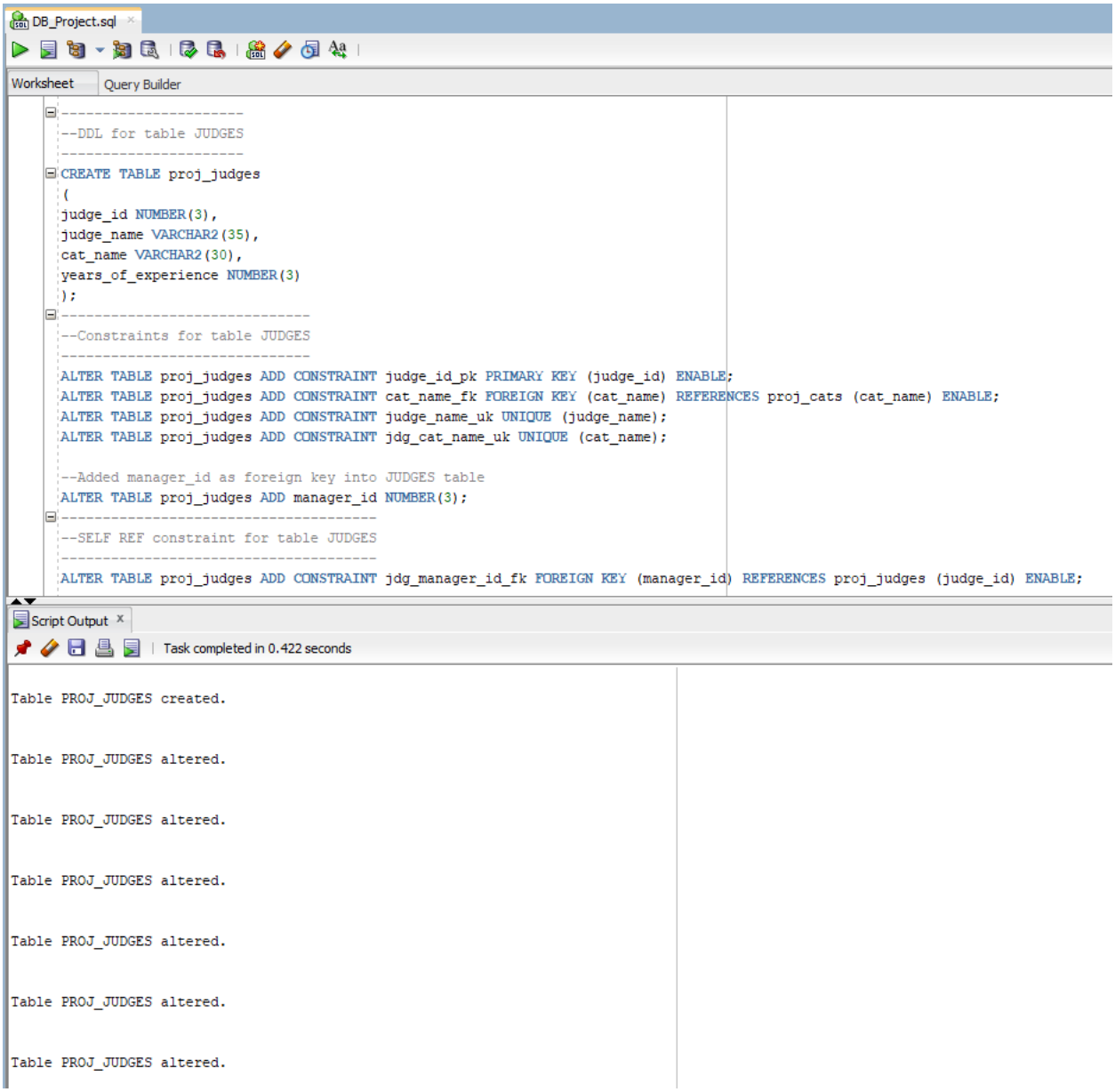

DB_Project.sql
Worksheet
Query Builder

--DDL for table PLACES

CREATE TABLE proj_places
(
final_placement VARCHAR2(6),
grade NUMBER(4,2),
cat_id NUMBER(3)
);

--Constraints for table PLACES

ALTER TABLE proj_places ADD CONSTRAINT place_pk PRIMARY KEY (final_placement) ENABLE;
ALTER TABLE proj_places ADD CONSTRAINT cat_id_fk FOREIGN KEY (cat_id) REFERENCES proj_cats (cat_id) ENABLE;
ALTER TABLE proj_places ADD CONSTRAINT cat_id_uk UNIQUE (cat_id);
Script Output
Task completed in 0.235 seconds
Table PROJ_PLACES created.
Table PROJ_PLACES altered.
Table PROJ_PLACES altered.
Table PROJ_PLACES altered.


DB_Project.sql
Worksheet
Query Builder

--DDL for table JUDGES

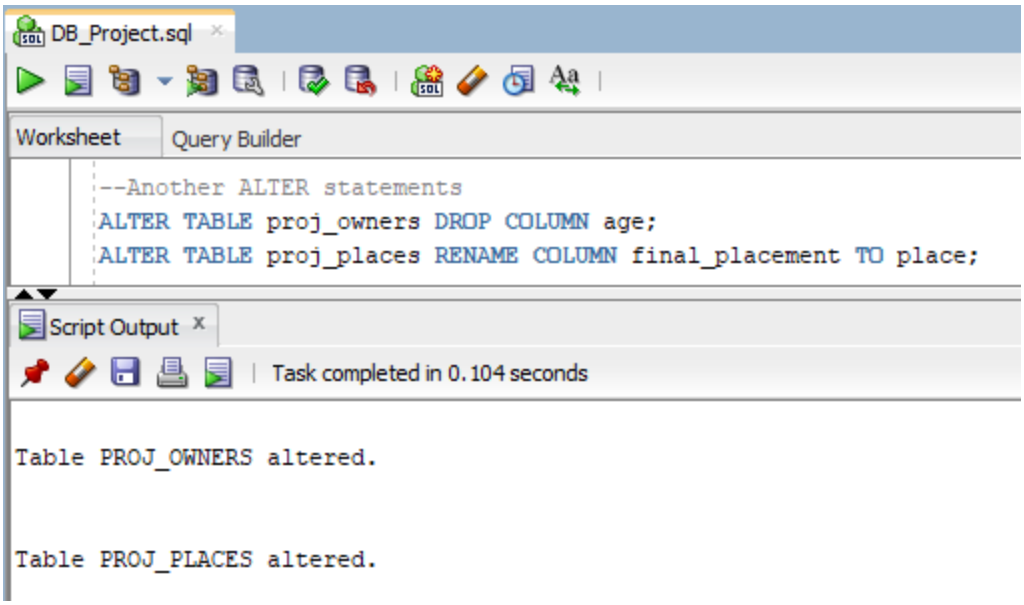
CREATE TABLE proj_judges
(
judge_id NUMBER(3),
judge_name VARCHAR2(35),
cat_name VARCHAR2(30),
years_of_experience NUMBER(3)
);

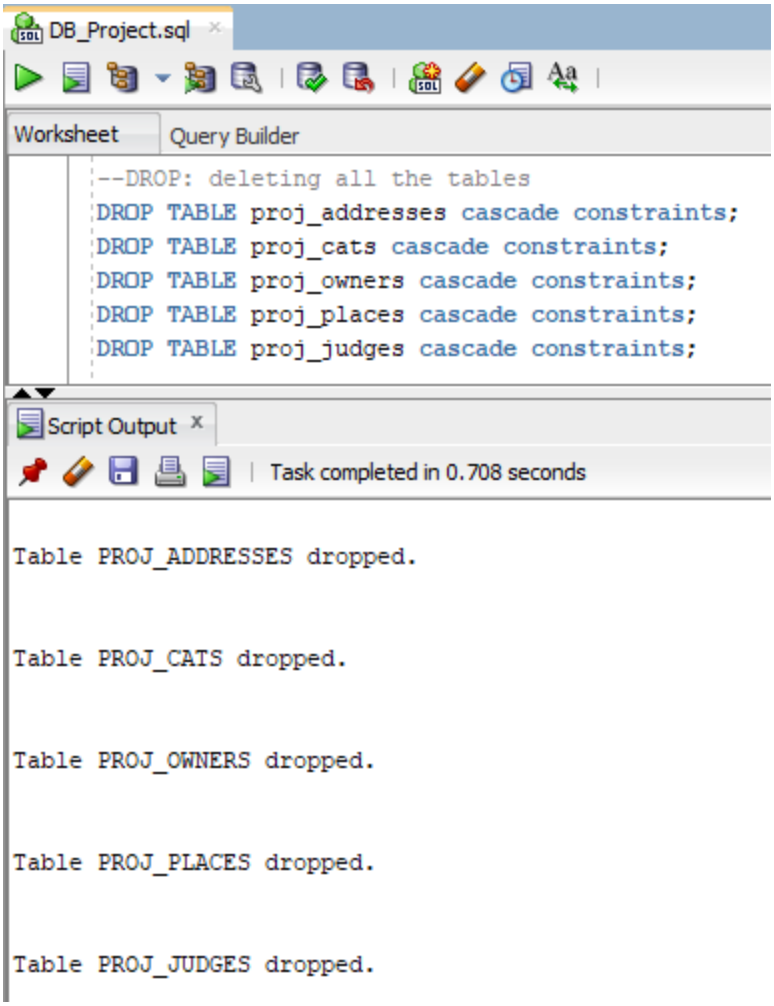
--Constraints for table JUDGES

ALTER TABLE proj_judges ADD CONSTRAINT judge_id_pk PRIMARY KEY (judge_id) ENABLE;
ALTER TABLE proj_judges ADD CONSTRAINT cat_name_fk FOREIGN KEY (cat_name) REFERENCES proj_cats (cat_name) ENABLE;
ALTER TABLE proj_judges ADD CONSTRAINT judge_name_uk UNIQUE (judge_name);
ALTER TABLE proj_judges ADD CONSTRAINT jdg_cat_name_uk UNIQUE (cat_name);
--Added manager_id as foreign key into JUDGES table
ALTER TABLE proj_judges ADD manager_id NUMBER(3);

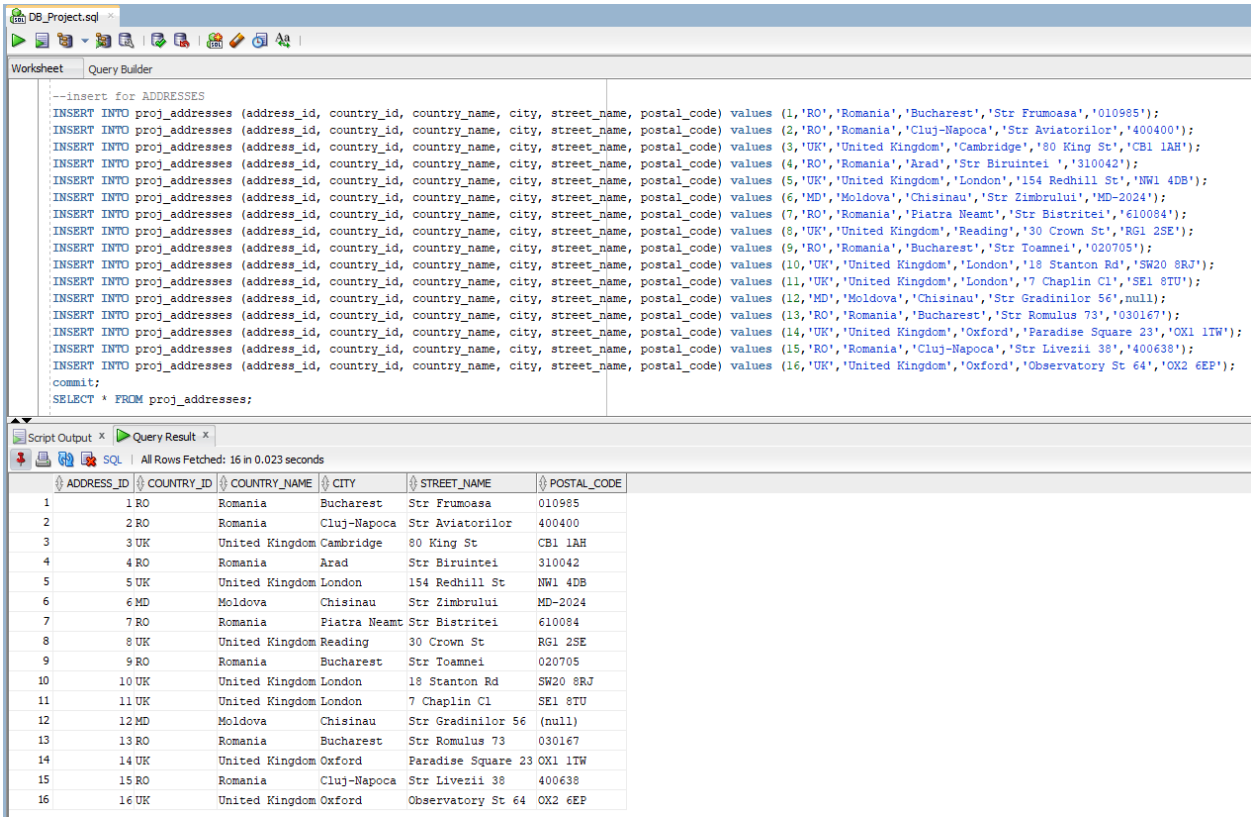
--SELF REF constraint for table JUDGES

ALTER TABLE proj_judges ADD CONSTRAINT jdg_manager_id_fk FOREIGN KEY (manager_id) REFERENCES proj_judges (judge_id) ENABLE;
Script Output
Task completed in 0.422 seconds
Table PROJ_JUDGES created.
Table PROJ_JUDGES altered.
Table PROJ_JUDGES altered.
Table PROJ_JUDGES altered.
Table PROJ_JUDGES altered.
Table PROJ_JUDGES altered.
Table PROJ_JUDGES altered.


DB_Project.sql
Worksheet
Query Builder
--Another ALTER statements
ALTER TABLE proj_owners DROP COLUMN age;
ALTER TABLE proj_places RENAME COLUMN final_placement TO place;
Script Output
Task completed in 0.104 seconds
Table PROJ_OWNERS altered.
Table PROJ_PLACES altered.


DB_Project.sql
Worksheet
Query Builder
--DROP: deleting all the tables
DROP TABLE proj_addresses cascade constraints;
DROP TABLE proj_cats cascade constraints;
DROP TABLE proj_owners cascade constraints;
DROP TABLE proj_places cascade constraints;
DROP TABLE proj_judges cascade constraints;
Script Output
Task completed in 0.708 seconds
Table PROJ_ADDRESSES dropped.
Table PROJ_CATS dropped.
Table PROJ_OWNERS dropped.
Table PROJ_PLACES dropped.
Table PROJ_JUDGES dropped.

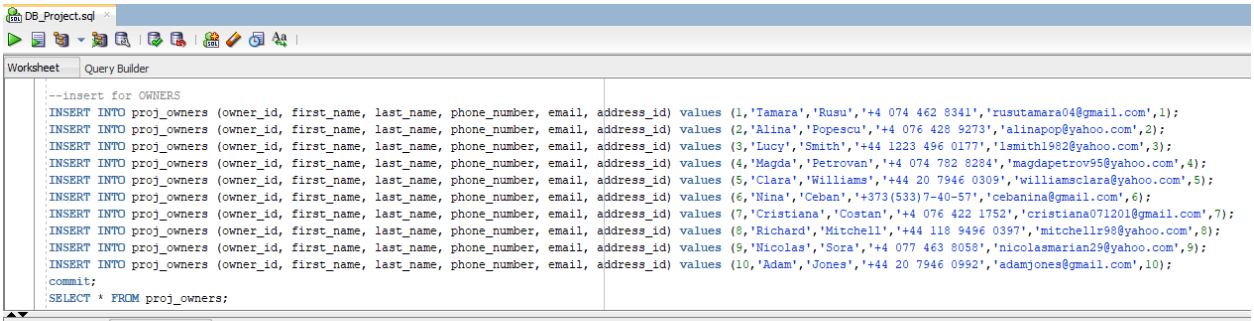
USING DML STATEMENTS



```
--insert for ADDRESSES
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (1,'RO','Romania','Bucharest','Str Frumoasa','010985');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (2,'RO','Romania','Cluj-Napoca','Str Aviatorilor','400400');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (3,'UK','United Kingdom','Cambridge','80 King St','CB1 1AH');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (4,'RO','Romania','Arad','Str Biruintei ','310042');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (5,'UK','United Kingdom','London','154 Redhill St','NW1 4DB');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (6,'MD','Moldova','Chisinau','Str Zimbrului','MD-2024');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (7,'RO','Romania','Piatra Neamt','Str Bistritei','610084');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (8,'UK','United Kingdom','Reading','30 Crown St','RG1 2SE');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (9,'RO','Romania','Bucharest','Str Toamnei','020705');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (10,'UK','United Kingdom','London','18 Stanton Rd','SW20 8RJ');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (11,'UK','United Kingdom','London','7 Chaplin Cl','SE1 8TU');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (12,'MD','Moldova','Chisinau','Str Gradinilor 56',null);
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (13,'RO','Romania','Bucharest','Str Romulus 73','030167');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (14,'UK','United Kingdom','Oxford','Paradise Square 23','OX1 1TW');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (15,'RO','Romania','Cluj-Napoca','Str Livezii 38','400638');
INSERT INTO proj_addresses (address_id, country_id, country_name, city, street_name, postal_code) values (16,'UK','United Kingdom','Oxford','Observatory St 64','OX2 6EP');
commit;
SELECT * FROM proj_addresses;
```

All Rows Fetched: 16 in 0.023 seconds

ADDRESS_ID	COUNTRY_ID	COUNTRY_NAME	CITY	STREET_NAME	POSTAL_CODE
1	RO	Romania	Bucharest	Str Frumoasa	010985
2	RO	Romania	Cluj-Napoca	Str Aviatorilor	400400
3	UK	United Kingdom	Cambridge	80 King St	CB1 1AH
4	RO	Romania	Arad	Str Biruintei	310042
5	UK	United Kingdom	London	154 Redhill St	NW1 4DB
6	MD	Moldova	Chisinau	Str Zimbrului	MD-2024
7	RO	Romania	Piatra Neamt	Str Bistritei	610084
8	UK	United Kingdom	Reading	30 Crown St	RG1 2SE
9	RO	Romania	Bucharest	Str Toamnei	020705
10	UK	United Kingdom	London	18 Stanton Rd	SW20 8RJ
11	UK	United Kingdom	London	7 Chaplin Cl	SE1 8TU
12	MD	Moldova	Chisinau	Str Gradinilor 56	(null)
13	RO	Romania	Bucharest	Str Romulus 73	030167
14	UK	United Kingdom	Oxford	Paradise Square 23	OX1 1TW
15	RO	Romania	Cluj-Napoca	Str Livezii 38	400638
16	UK	United Kingdom	Oxford	Observatory St 64	OX2 6EP



```
--insert for OWNERS
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (1,'Tamara','Rusu','+4 074 462 8341','rusutamara04@gmail.com',1);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (2,'Alina','Popescu','+4 076 428 9273','alinapop@yahoo.com',2);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (3,'Lucy','Smith','+44 1223 496 0177','lsmith1982@yahoo.com',3);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (4,'Magda','Petrovan','+4 074 782 8284','magdapetrov95@yahoo.com',4);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (5,'Clara','Williams','+44 20 7946 0309','williamsclara@yahoo.com',5);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (6,'Nina','Ceban','+373(533)7-40-57','cebanina@gmail.com',6);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (7,'Cristiana','Costan','+4 076 422 1752','cristiana071201@gmail.com',7);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (8,'Richard','Mitchell','+44 118 9496 0397','mitchellr98@yahoo.com',8);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (9,'Nicolas','Sora','+4 077 463 8058','nicolasmarian29@yahoo.com',9);
INSERT INTO proj_owners (owner_id, first_name, last_name, phone_number, email, address_id) values (10,'Adam','Jones','+44 20 7946 0992','adamjones@gmail.com',10);
commit;
SELECT * FROM proj_owners;
```

All Rows Fetched: 10 in 0.022 seconds

OWNER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ADDRESS_ID
1	Tamara	Rusu	+4 074 462 8341	rusutamara04@gmail.com	1
2	Alina	Popescu	+4 076 428 9273	alinapop@yahoo.com	2
3	Lucy	Smith	+44 1223 496 0177	lsmith1982@yahoo.com	3
4	Magda	Petrovan	+4 074 782 8284	magdapetrov95@yahoo.com	4
5	Clara	Williams	+44 20 7946 0309	williamsclara@yahoo.com	5
6	Nina	Ceban	+373(533)7-40-57	cebanina@gmail.com	6
7	Cristiana	Costan	+4 076 422 1752	cristiana071201@gmail.com	7
8	Richard	Mitchell	+44 118 9496 0397	mitchellr98@yahoo.com	8
9	Nicolas	Sora	+4 077 463 8058	nicolasmarian29@yahoo.com	9
10	Adam	Jones	+44 20 7946 0992	adamjones@gmail.com	10

OWNER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ADDRESS_ID
7	Cristiana	Costan	+4 076 422 1752	cristiana071201@gmail.com	7

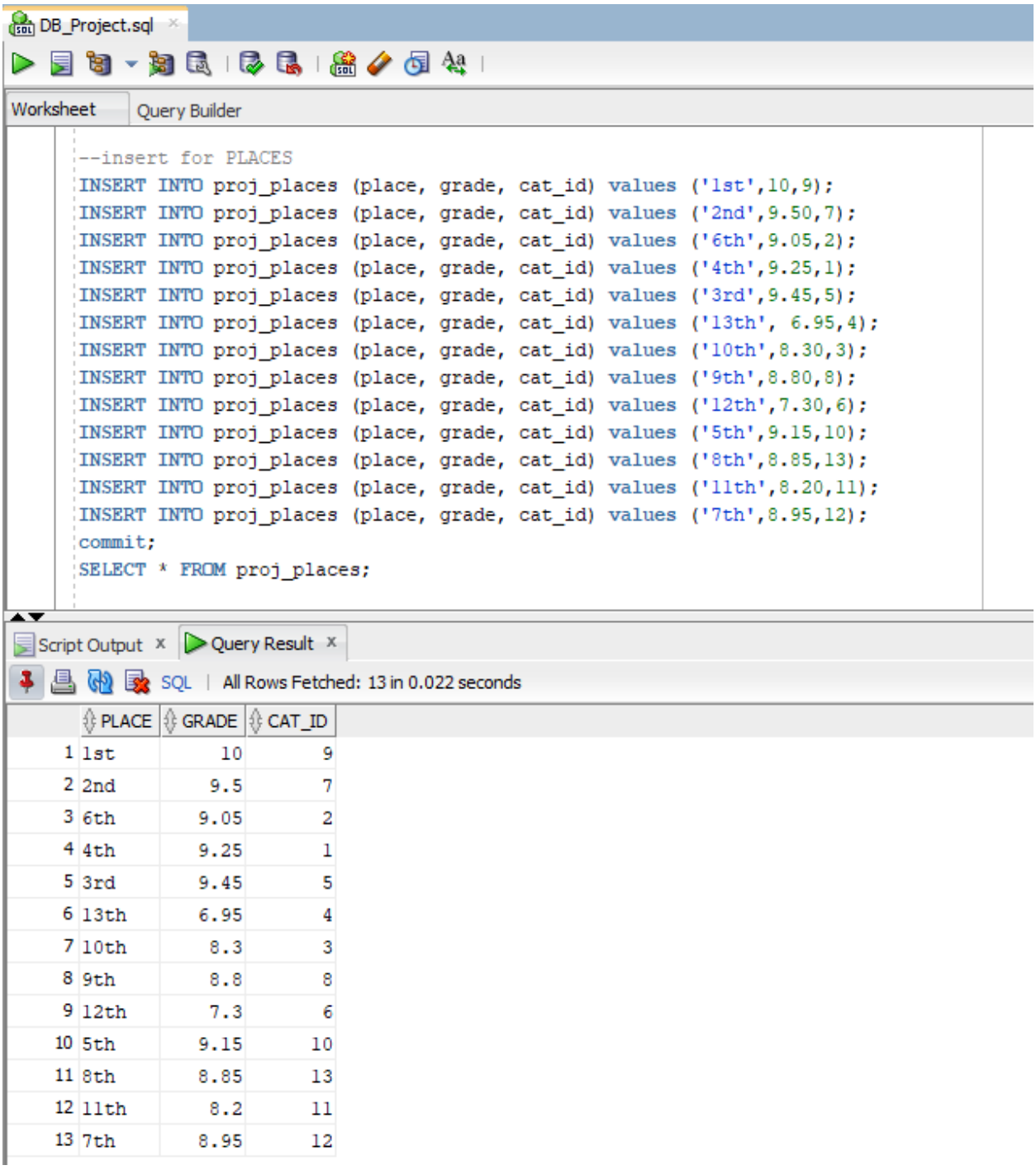
DB_Project.sql

Worksheet | Query Builder

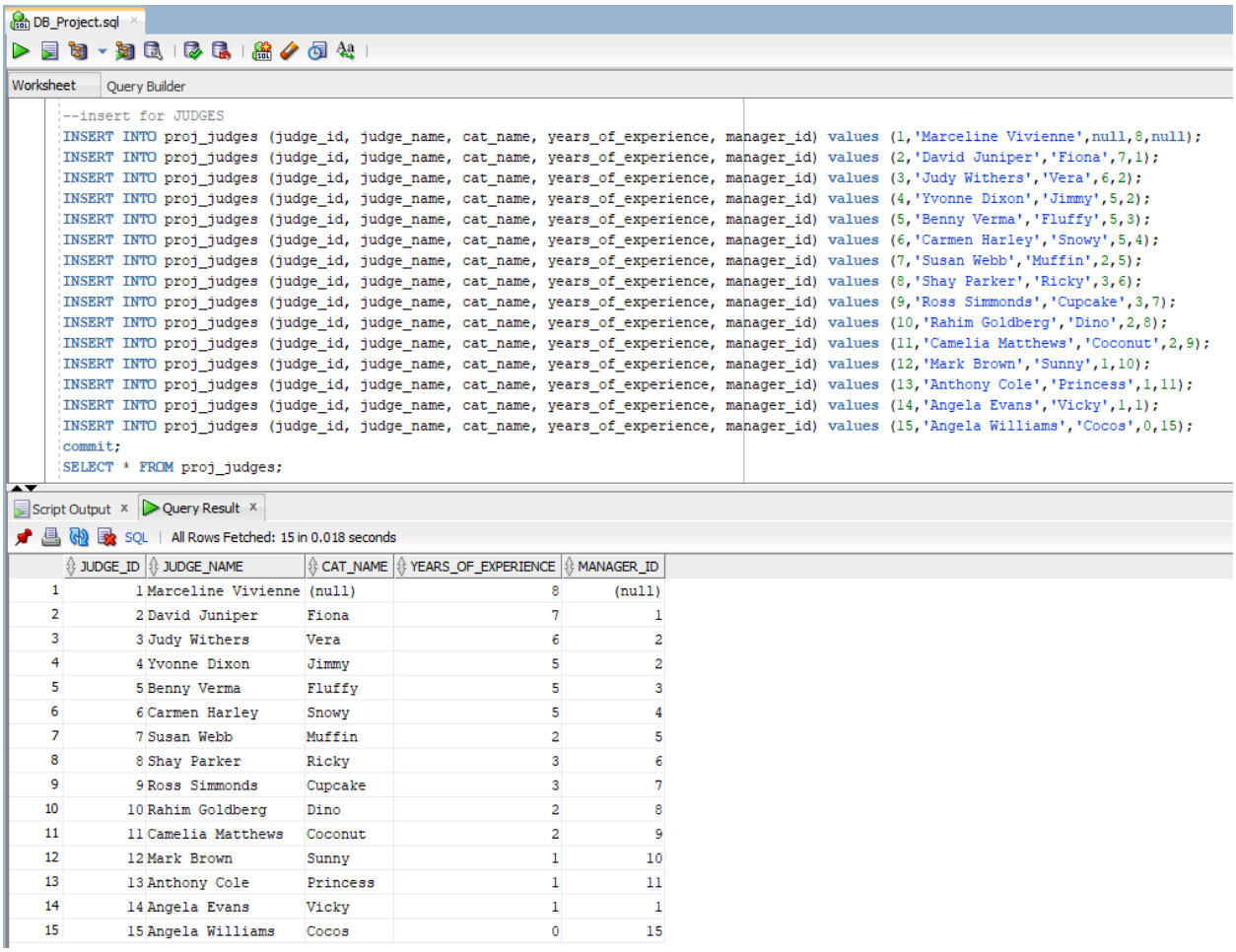
```
--insert for CATS
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (1,'Vera','Sphynx',1,to_date('14-05-2021','DD-MM-RRRR'),'F','medium','none','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (2,'Snowy','Maine Coon',2,to_date('21-06-2018','DD-MM-RRRR'),'M','large','long','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (3,'Sunny','Maine Coon',2,to_date('21-06-2019','DD-MM-RRRR'),'F','large','long','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (4,'Cupcake','Munchkin',3,to_date('15-12-2020','DD-MM-RRRR'),'F','small','medium','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (5,'Princess','Birman',4,to_date('28-02-2020','DD-MM-RRRR'),'F','medium','long','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (6,'Fluffy','Birman',4,to_date('27-07-2021','DD-MM-RRRR'),'M','medium','long','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (7,'Jimmy','Sphynx',5,to_date('25-10-2019','DD-MM-RRRR'),'M','medium','none','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (8,'Dino','Devon Rex',6,to_date('19-11-2017','DD-MM-RRRR'),'M','medium','short','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (9,'Muffin','Ragamuffin',7,to_date('07-12-2021','DD-MM-RRRR'),'F','medium','long','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (10,'Vicky','British Shorthair',8,to_date('04-01-2020','DD-MM-RRRR'),'F','medium','medium','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (11,'Ricky','British Shorthair',8,to_date('04-01-2020','DD-MM-RRRR'),'M','medium','medium','YES');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (12,'Fiona','Munchkin',9,to_date('09-03-2017','DD-MM-RRRR'),'F','small','medium','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (13,'Coconut','Maine Coon',10,to_date('11-07-2017','DD-MM-RRRR'),'F','large','long','NO');
INSERT INTO proj_cats (cat_id, cat_name, breed, owner_id, birth_date, sex, cat_size, fur_type, other_contests) values (14,'Cocos','Maine Coon',10,to_date('17-03-2016','DD-MM-RRRR'),'M','large','long','YES');
commit;
SELECT * FROM proj_cats;
```

Script Output | Query Result | SQL | All Rows Fetched: 14 in 0.018 seconds

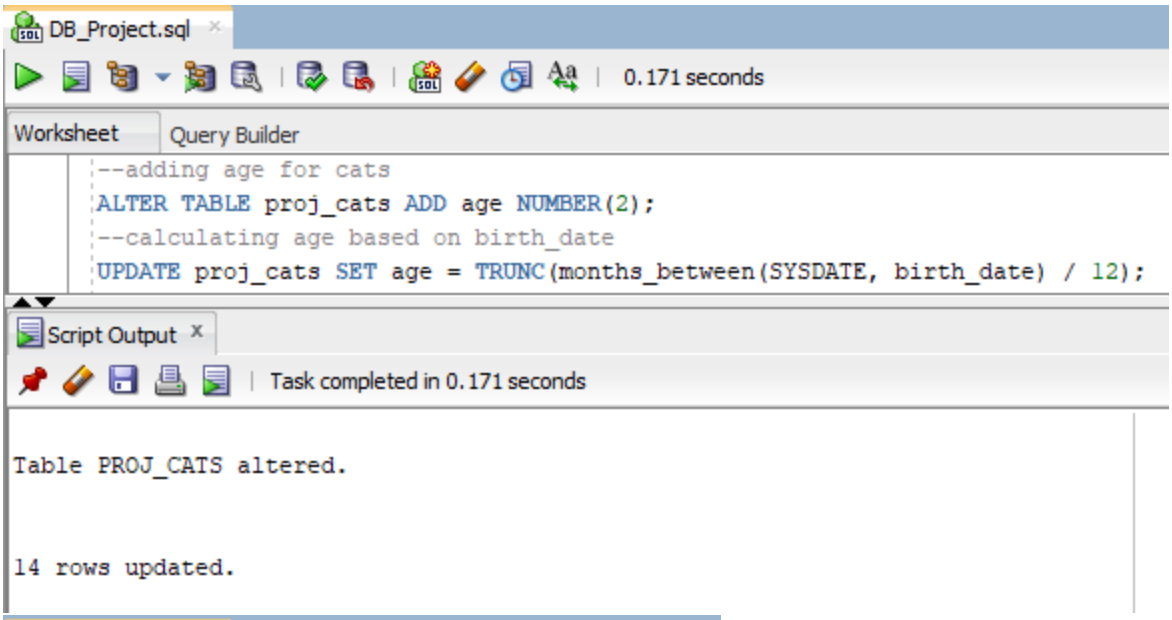
	CAT_ID	CAT_NAME	BREED	OWNER_ID	BIRTH_DATE	SEX	CAT_SIZE	FUR_TYPE	OTHER_CONTESTS
1	1	Vera	Sphynx	1	14-MAY-21	F	medium	none	YES
2	2	Snowy	Maine Coon	2	21-JUN-18	M	large	long	NO
3	3	Sunny	Maine Coon	2	21-JUN-19	F	large	long	NO
4	4	Cupcake	Munchkin	3	15-DEC-20	F	small	medium	YES
5	5	Princess	Birman	4	28-FEB-20	F	medium	long	YES
6	6	Fluffy	Birman	4	27-JUL-21	M	medium	long	NO
7	7	Jimmy	Sphynx	5	25-OCT-19	M	medium	none	YES
8	8	Dino	Devon Rex	6	19-NOV-17	M	medium	short	NO
9	9	Muffin	Ragamuffin	7	07-DEC-21	F	medium	long	NO
10	10	Vicky	British Shorthair	8	04-JAN-20	F	medium	medium	YES
11	11	Ricky	British Shorthair	8	04-JAN-20	M	medium	medium	YES
12	12	Fiona	Munchkin	9	09-MAR-17	F	small	medium	NO
13	13	Coconut	Maine Coon	10	11-JUL-17	F	large	long	NO
14	14	Cocos	Maine Coon	10	17-MAR-16	M	large	long	YES

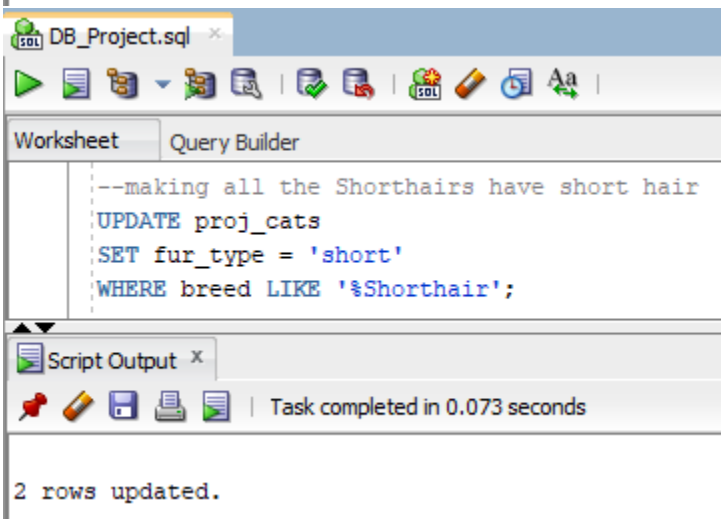


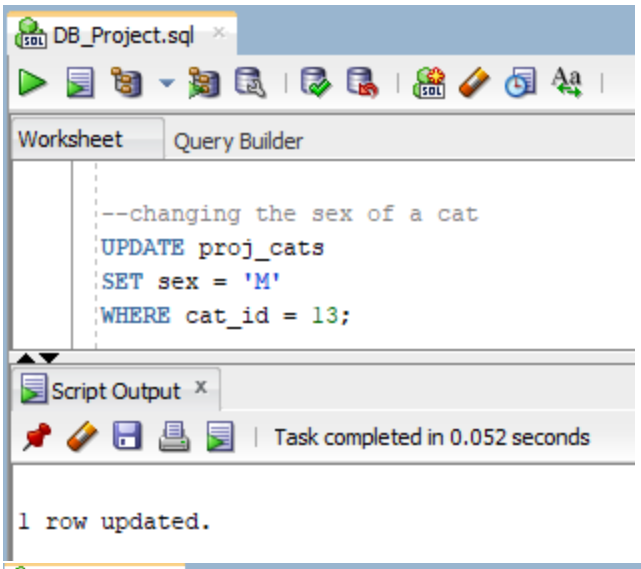
	PLACE	GRADE	CAT_ID
1	1st	10	9
2	2nd	9.5	7
3	6th	9.05	2
4	4th	9.25	1
5	3rd	9.45	5
6	13th	6.95	4
7	10th	8.3	3
8	9th	8.8	8
9	12th	7.3	6
10	5th	9.15	10
11	8th	8.85	13
12	11th	8.2	11
13	7th	8.95	12



	JUDGE_ID	JUDGE_NAME	CAT_NAME	YEARS_OF_EXPERIENCE	MANAGER_ID
1	1	Marceline Vivienne	(null)	8	(null)
2	2	David Juniper	Fiona	7	1
3	3	Judy Withers	Vera	6	2
4	4	Yvonne Dixon	Jimmy	5	2
5	5	Benny Verma	Fluffy	5	3
6	6	Carmen Harley	Snowy	5	4
7	7	Susan Webb	Muffin	2	5
8	8	Shay Parker	Ricky	3	6
9	9	Ross Simmonds	Cupcake	3	7
10	10	Rahim Goldberg	Dino	2	8
11	11	Camelia Matthews	Coconut	2	9
12	12	Mark Brown	Sunny	1	10
13	13	Anthony Cole	Princess	1	11
14	14	Angela Evans	Vicky	1	1
15	15	Angela Williams	Cocos	0	15


DB_Project.sql
0.171 seconds
Worksheet
Query Builder
--adding age for cats
ALTER TABLE proj_cats ADD age NUMBER(2);
--calculating age based on birth_date
UPDATE proj_cats SET age = TRUNC(months_between(SYSDATE, birth_date) / 12);
Script Output
Task completed in 0.171 seconds
Table PROJ_CATS altered.
14 rows updated.


DB_Project.sql
Worksheet
Query Builder
--making all the Shorthairs have short hair
UPDATE proj_cats
SET fur_type = 'short'
WHERE breed LIKE '%Shorthair';
Script Output
Task completed in 0.073 seconds
2 rows updated.



DB_Project.sql

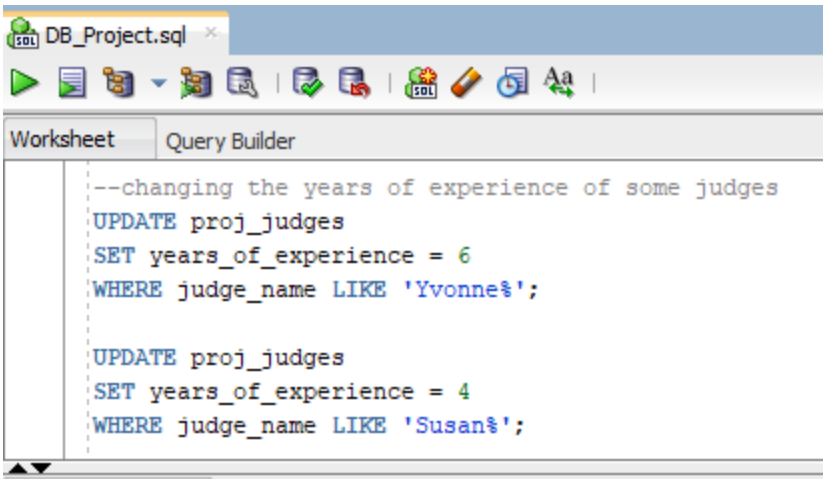
Worksheet | Query Builder

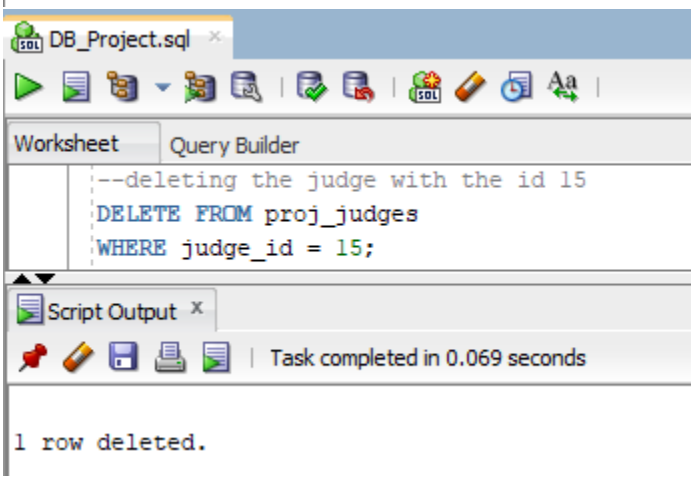
```
SELECT * FROM proj_cats;
```

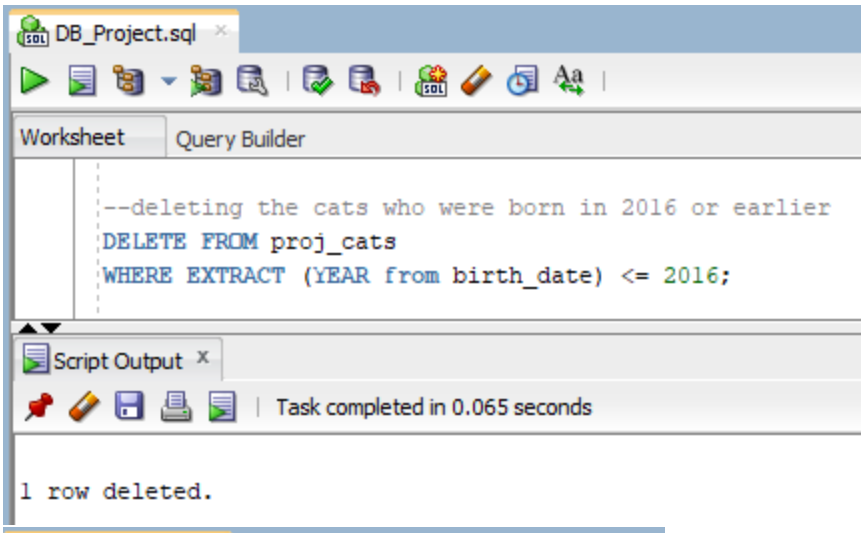
Script Output | Query Result

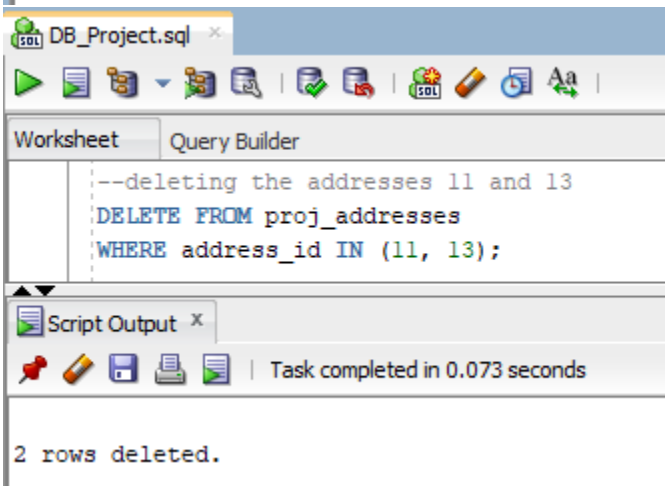
SQL | All Rows Fetched: 14 in 0.02 seconds

	CAT_ID	CAT_NAME	BREED	OWNER_ID	BIRTH_DATE	SEX	CAT_SIZE	FUR_TYPE	OTHER_CONTESTS	AGE
1	1	Vera	Sphynx	1	14-MAY-21	F	medium	none	YES	1
2	2	Snowy	Maine Coon	2	21-JUN-18	M	large	long	NO	4
3	3	Sunny	Maine Coon	2	21-JUN-19	F	large	long	NO	3
4	4	Cupcake	Munchkin	3	15-DEC-20	F	small	medium	YES	2
5	5	Princess	Birman	4	28-FEB-20	F	medium	long	YES	2
6	6	Fluffy	Birman	4	27-JUL-21	M	medium	long	NO	1
7	7	Jimmy	Sphynx	5	25-OCT-19	M	medium	none	YES	3
8	8	Dino	Devon Rex	6	19-NOV-17	M	medium	short	NO	5
9	9	Muffin	Ragamuffin	7	07-DEC-21	F	medium	long	NO	1
10	10	Vicky	British Shorthair	8	04-JAN-20	F	medium	short	YES	2
11	11	Ricky	British Shorthair	8	04-JAN-20	M	medium	short	YES	2
12	12	Fiona	Munchkin	9	09-MAR-17	F	small	medium	NO	5
13	13	Coconut	Maine Coon	10	11-JUL-17	M	large	long	NO	5
14	14	Cocos	Maine Coon	10	17-MAR-16	M	large	long	YES	6

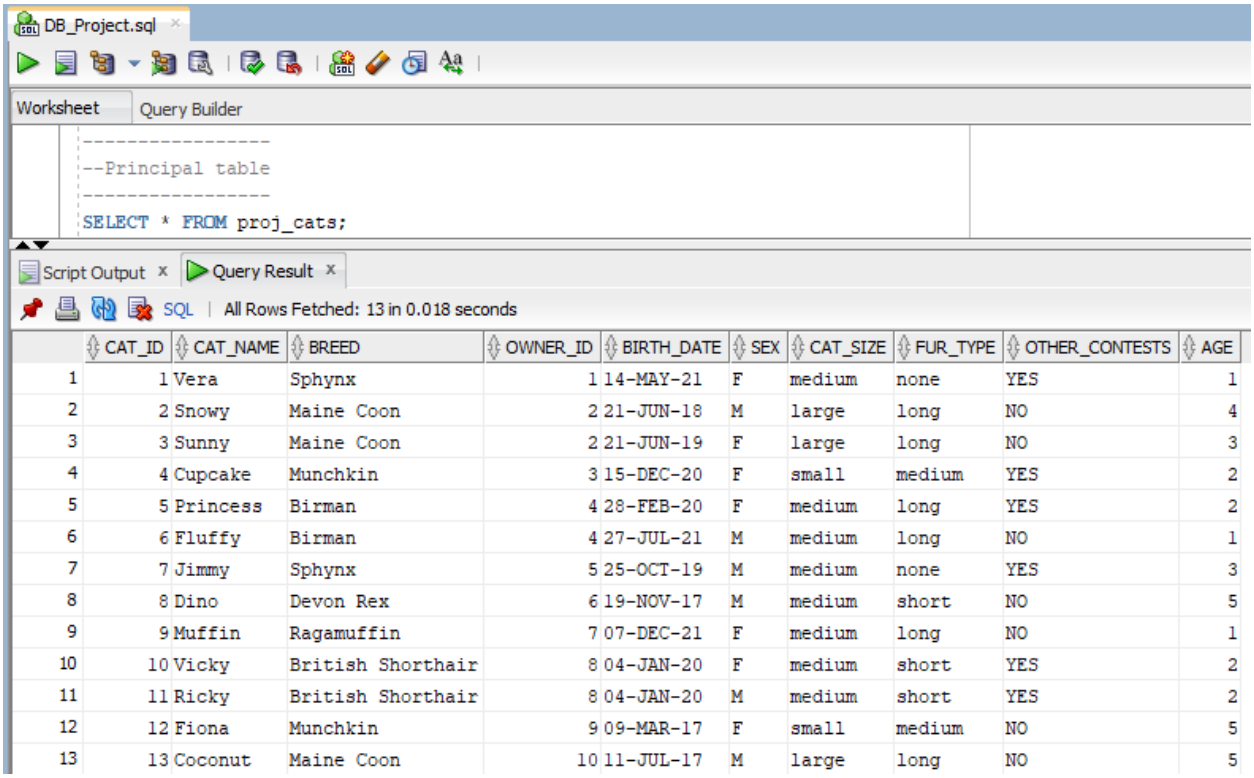

DB_Project.sql
Worksheet
Query Builder
--changing the years of experience of some judges
UPDATE proj_judges
SET years_of_experience = 6
WHERE judge_name LIKE 'Yvonne%';
UPDATE proj_judges
SET years_of_experience = 4
WHERE judge_name LIKE 'Susan%';
Script Output
Task completed in 0.081 seconds
1 row updated.
1 row updated.


DB_Project.sql
Worksheet
Query Builder
--deleting the judge with the id 15
DELETE FROM proj_judges
WHERE judge_id = 15;
Script Output
Task completed in 0.069 seconds
1 row deleted.

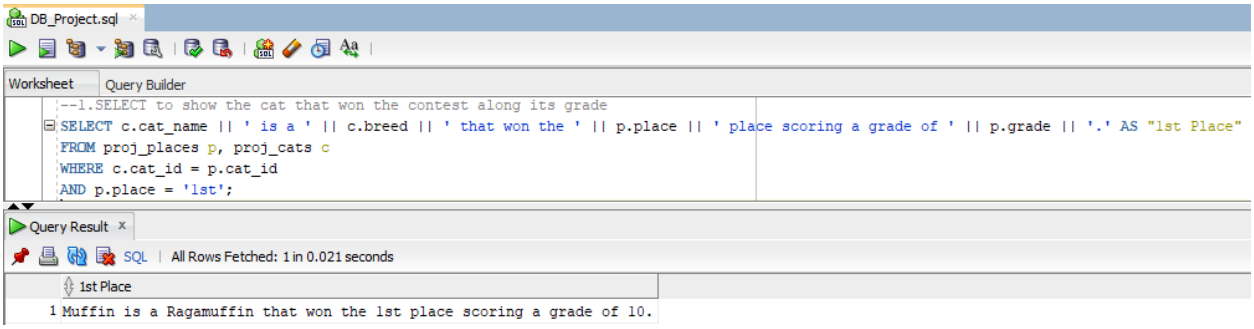

DB_Project.sql
Worksheet
Query Builder
--deleting the cats who were born in 2016 or earlier
DELETE FROM proj_cats
WHERE EXTRACT (YEAR from birth_date) <= 2016;
Script Output
Task completed in 0.065 seconds
1 row deleted.


DB_Project.sql
Worksheet
Query Builder
--deleting the addresses 11 and 13
DELETE FROM proj_addresses
WHERE address_id IN (11, 13);
Script Output
Task completed in 0.073 seconds
2 rows deleted.

DIVERSE AND RELEVANT SELECT STATEMENTS

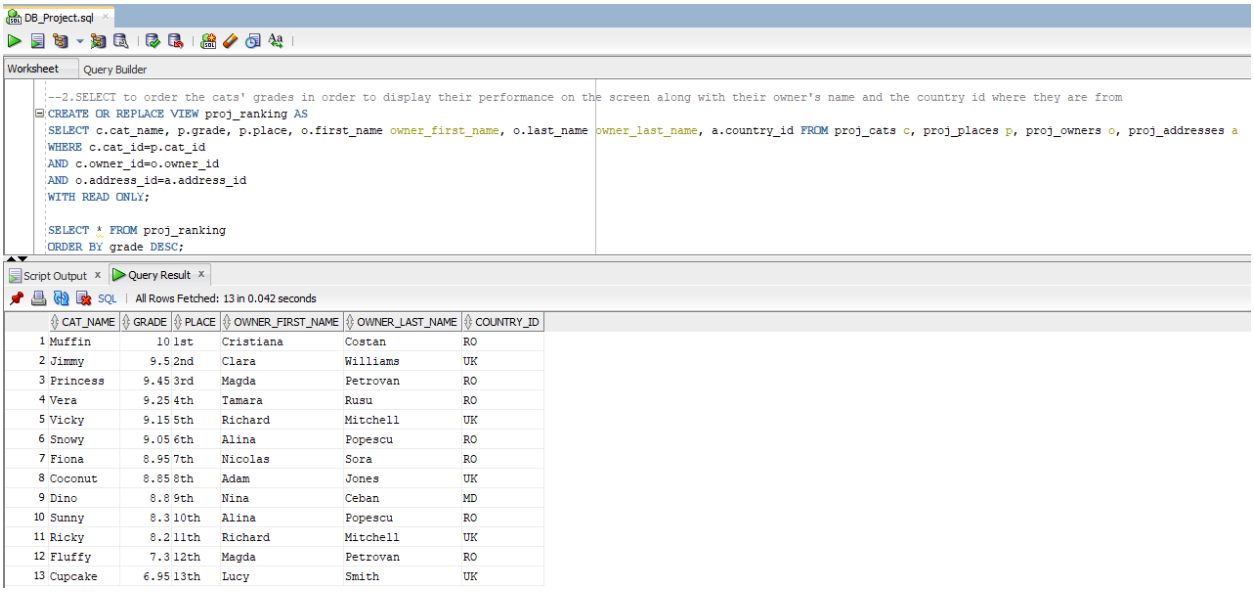


	CAT_ID	CAT_NAME	BREED	OWNER_ID	BIRTH_DATE	SEX	CAT_SIZE	FUR_TYPE	OTHER_CONTESTS	AGE
1	1	Vera	Sphynx	1	14-MAY-21	F	medium	none	YES	1
2	2	Snowy	Maine Coon	2	21-JUN-18	M	large	long	NO	4
3	3	Sunny	Maine Coon	2	21-JUN-19	F	large	long	NO	3
4	4	Cupcake	Munchkin	3	15-DEC-20	F	small	medium	YES	2
5	5	Princess	Birman	4	28-FEB-20	F	medium	long	YES	2
6	6	Fluffy	Birman	4	27-JUL-21	M	medium	long	NO	1
7	7	Jimmy	Sphynx	5	25-OCT-19	M	medium	none	YES	3
8	8	Dino	Devon Rex	6	19-NOV-17	M	medium	short	NO	5
9	9	Muffin	Ragamuffin	7	07-DEC-21	F	medium	long	NO	1
10	10	Vicky	British Shorthair	8	04-JAN-20	F	medium	short	YES	2
11	11	Ricky	British Shorthair	8	04-JAN-20	M	medium	short	YES	2
12	12	Fiona	Munchkin	9	09-MAR-17	F	small	medium	NO	5
13	13	Coconut	Maine Coon	10	11-JUL-17	M	large	long	NO	5



	1st Place
1	Muffin is a Ragamuffin that won the 1st place scoring a grade of 10.

For performance reasons, I avoided using “ORDER BY” when creating a view. Instead, I used it for displaying the results.



DB_Project.sql

Worksheet | Query Builder

```
--2.SELECT to order the cats' grades in order to display their performance on the screen along with their owner's name and the country id where they are from
CREATE OR REPLACE VIEW proj_ranking AS
SELECT c.cat_name, p.grade, p.place, o.first_name owner_first_name, o.last_name owner_last_name, a.country_id FROM proj_cats c, proj_places p, proj_owners o, proj_addresses a
WHERE c.cat_id=p.cat_id
AND c.owner_id=o.owner_id
AND o.address_id=a.address_id
WITH READ ONLY;

SELECT * FROM proj_ranking
ORDER BY grade DESC;
```

Script Output | Query Result

All Rows Fetched: 13 in 0.042 seconds

	CAT_NAME	GRADE	PLACE	OWNER_FIRST_NAME	OWNER_LAST_NAME	COUNTRY_ID
1	Muffin	10	1st	Cristiana	Costan	RO
2	Jimmy	9.5	2nd	Clara	Williams	UK
3	Princess	9.45	3rd	Magda	Petrovan	RO
4	Vera	9.25	4th	Tamara	Rusu	RO
5	Vicky	9.15	5th	Richard	Mitchell	UK
6	Snowy	9.05	6th	Alina	Popescu	RO
7	Fiona	8.95	7th	Nicolas	Sora	RO
8	Coconut	8.85	8th	Adam	Jones	UK
9	Dino	8.8	9th	Nina	Ceban	MD
10	Sunny	8.3	10th	Alina	Popescu	RO
11	Ricky	8.2	11th	Richard	Mitchell	UK
12	Fluffy	7.3	12th	Magda	Petrovan	RO
13	Cupcake	6.95	13th	Lucy	Smith	UK

DB_Project.sql

Worksheet | Query Builder

```
--3.SELECT to see which owners have more than 1 cat in the contest
SELECT o.last_name, COUNT(*) owned_cats FROM proj_cats c, proj_owners o
WHERE c.owner_id = o.owner_id
HAVING COUNT(*) > 1
GROUP BY last_name
ORDER BY last_name;
```

Query Result

SQL | All Rows Fetched: 3 in 0.017 seconds

	LAST_NAME	OWNED_CATS
1	Mitchell	2
2	Petrovan	2
3	Popescu	2

DB_Project.sql

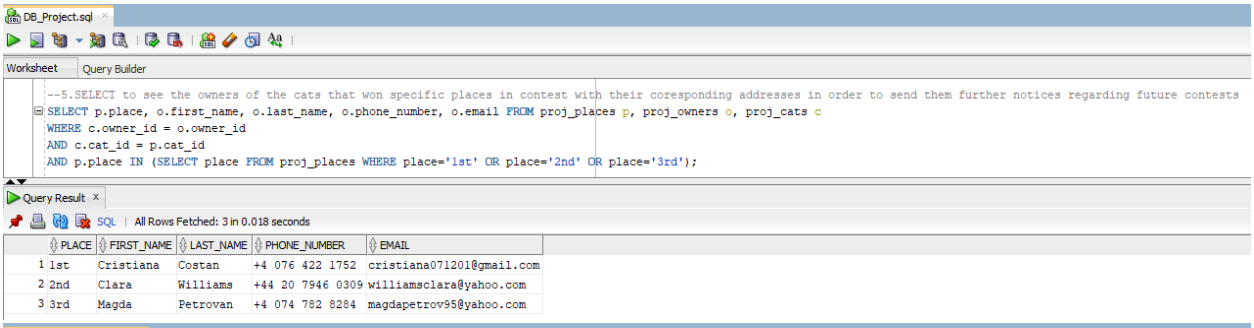
Worksheet | Query Builder

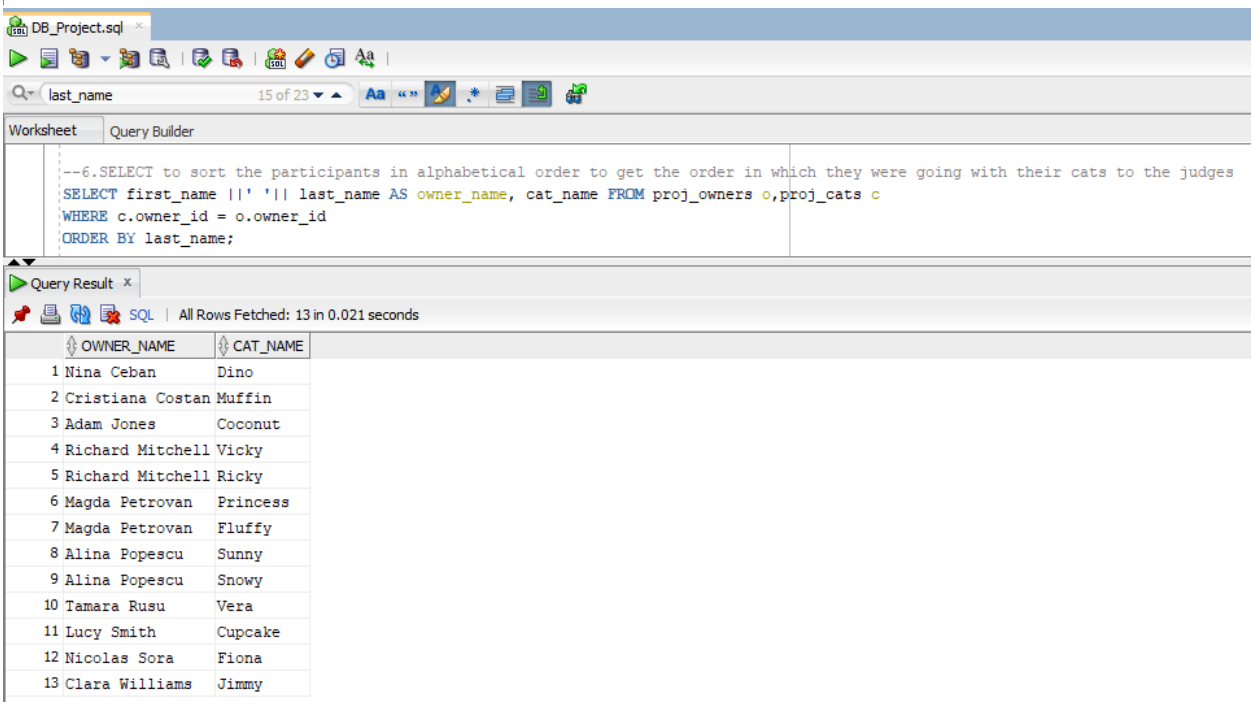
```
--4.SELECT to see which cats earn a Participation diploma and which cats earn a Winner diploma based on their places in contest
SELECT c.cat_name,p.grade,
CASE WHEN p.place NOT IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') THEN 'Participation'
WHEN p.place IN (SELECT place FROM proj_places WHERE place='1st') THEN 'Winner'
WHEN p.place IN (SELECT place FROM proj_places WHERE place='2nd') THEN 'Second place'
WHEN p.place IN (SELECT place FROM proj_places WHERE place='3rd') THEN 'Third place'
ELSE NULL
END diploma
FROM proj_places p, proj_cats c
WHERE p.cat_id = c.cat_id
ORDER BY p.grade DESC;
```

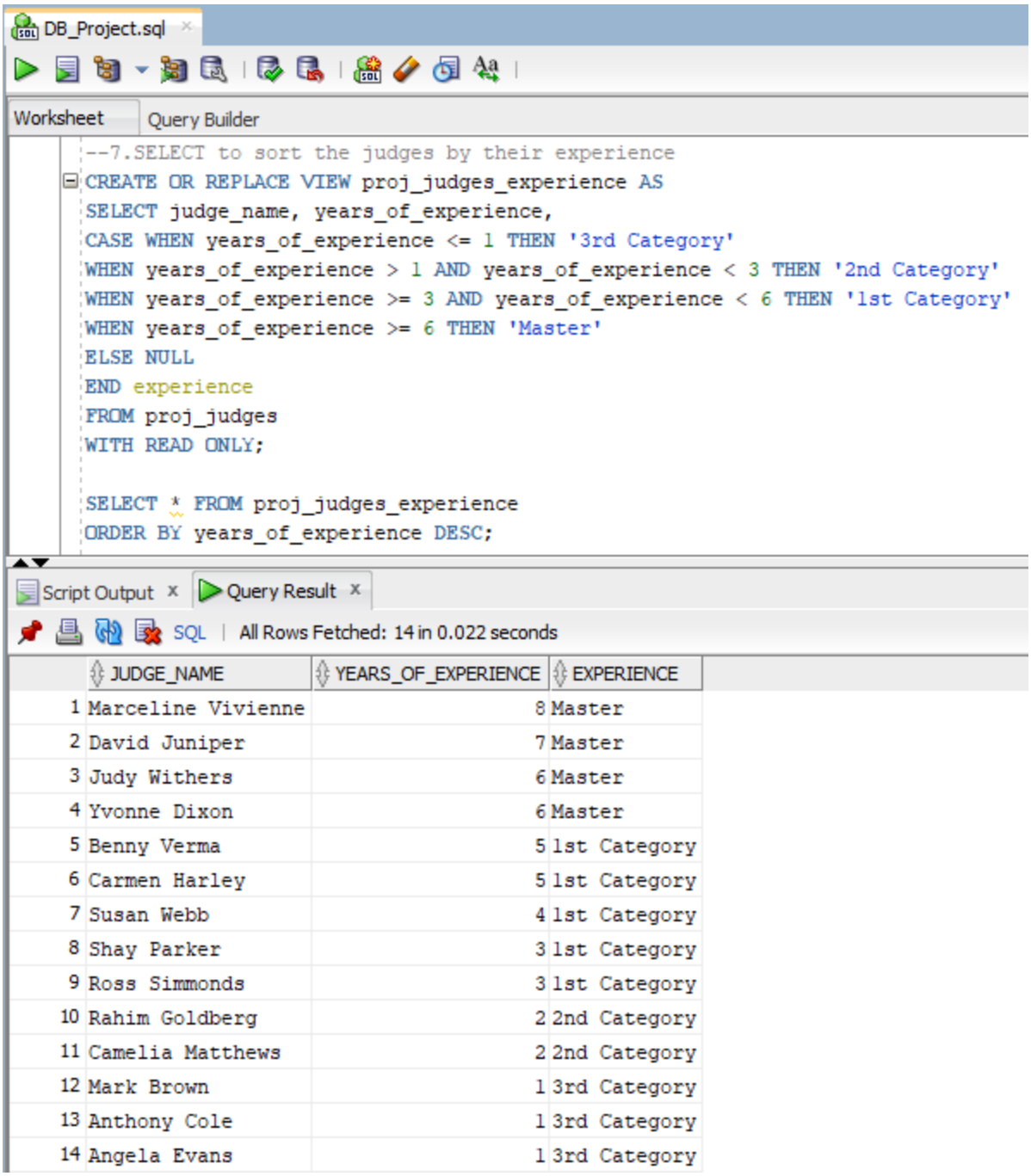
Query Result

SQL | All Rows Fetched: 13 in 0.02 seconds

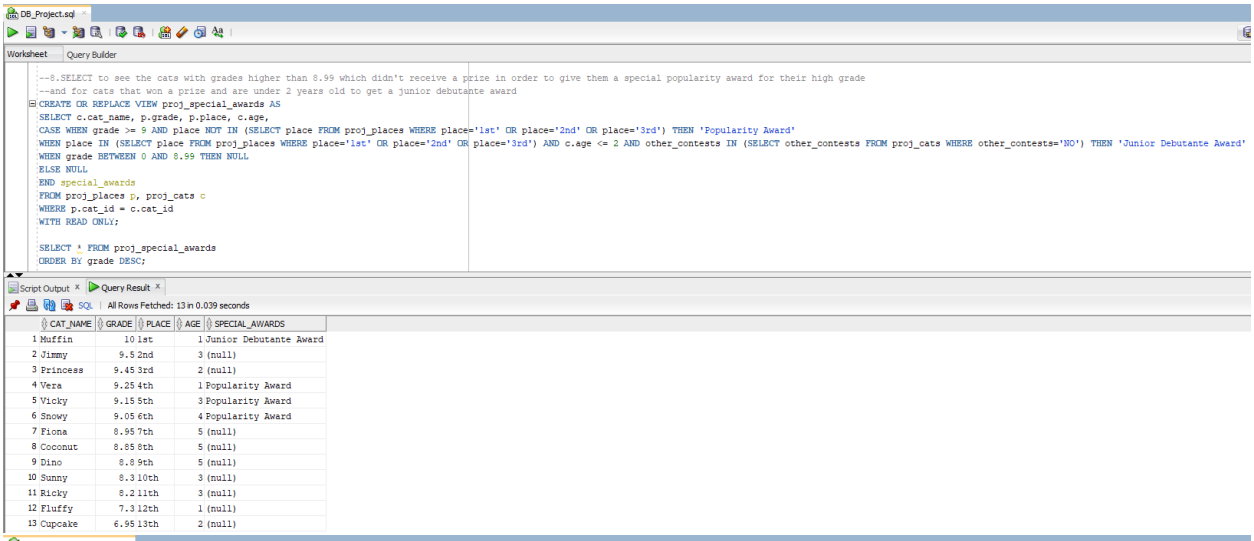
	CAT_NAME	GRADE	DIPLOMA
1	Muffin	10	Winner
2	Jimmy	9.5	Second place
3	Princess	9.45	Third place
4	Vera	9.25	Participation
5	Vicky	9.15	Participation
6	Snowy	9.05	Participation
7	Fiona	8.95	Participation
8	Coconut	8.85	Participation
9	Dino	8.8	Participation
10	Sunny	8.3	Participation
11	Ricky	8.2	Participation
12	Fluffy	7.3	Participation
13	Cupcake	6.95	Participation


DB_Project.sql
Worksheet
Query Builder
--5.SELECT to see the owners of the cats that won specific places in contest with their coresponding addresses in order to send them further notices regarding future contests
SELECT p.place, o.first_name, o.last_name, o.phone_number, o.email FROM proj_places p, proj_owners o, proj_cats c
WHERE c.owner_id = o.owner_id
AND c.cat_id = p.cat_id
AND p.place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd');
Query Result
SQL | All Rows Fetched: 3 in 0.018 seconds
PLACE FIRST_NAME LAST_NAME PHONE_NUMBER EMAIL
1 1st Cristiana Costan +4 076 422 1752 cristiana071201@gmail.com
2 2nd Clara Williams +44 20 7946 0309 williamsclara@yahoo.com
3 3rd Magda Petrovan +4 074 782 8284 magdapetrov95@yahoo.com


DB_Project.sql
last_name
15 of 23
Worksheet
Query Builder
--6.SELECT to sort the participants in alphabetical order to get the order in which they were going with their cats to the judges
SELECT first_name ||' '|| last_name AS owner_name, cat_name FROM proj_owners o,proj_cats c
WHERE c.owner_id = o.owner_id
ORDER BY last_name;
Query Result
SQL | All Rows Fetched: 13 in 0.021 seconds
OWNER_NAME CAT_NAME
1 Nina Ceban Dino
2 Cristiana Costan Muffin
3 Adam Jones Coconut
4 Richard Mitchell Vicky
5 Richard Mitchell Ricky
6 Magda Petrovan Princess
7 Magda Petrovan Fluffy
8 Alina Popescu Sunny
9 Alina Popescu Snowy
10 Tamara Rusu Vera
11 Lucy Smith Cupcake
12 Nicolas Sora Fiona
13 Clara Williams Jimmy



	JUDGE_NAME	YEARS_OF_EXPERIENCE	EXPERIENCE
1	Marceline Vivienne	8	Master
2	David Juniper	7	Master
3	Judy Withers	6	Master
4	Yvonne Dixon	6	Master
5	Benny Verma	5	1st Category
6	Carmen Harley	5	1st Category
7	Susan Webb	4	1st Category
8	Shay Parker	3	1st Category
9	Ross Simmonds	3	1st Category
10	Rahim Goldberg	2	2nd Category
11	Camelia Matthews	2	2nd Category
12	Mark Brown	1	3rd Category
13	Anthony Cole	1	3rd Category
14	Angela Evans	1	3rd Category



```
--8.SELECT to see the cats with grades higher than 8.99 which didn't receive a prize in order to give them a special popularity award for their high grade
--and for cats that won a prize and are under 2 years old to get a junior debutante award
CREATE OR REPLACE VIEW proj_special_awards AS
SELECT c.cat_name, p.grade, p.place, c.age,
CASE WHEN grade >= 9 AND place NOT IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') THEN 'Popularity Award'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND c.age <= 2 AND other_contests IN (SELECT other_contests FROM proj_cats WHERE other_contests='NO') THEN 'Junior Debutante Award'
WHEN grade BETWEEN 0 AND 8.99 THEN NULL
ELSE NULL
END special_awards
FROM proj_places p, proj_cats c
WHERE p.cat_id = c.cat_id
WITH READ ONLY;

SELECT * FROM proj_special_awards
ORDER BY grade DESC;
```

Script Output | Query Result

SQL | All Rows Fetched: 13 in 0.039 seconds

	CAT_NAME	GRADE	PLACE	AGE	SPECIAL_AWARDS
1	Muffin	10	1st	1	Junior Debutante Award
2	Jimmy	9.5	2nd	3	(null)
3	Princess	9.45	3rd	2	(null)
4	Vera	9.25	4th	1	Popularity Award
5	Vicky	9.15	5th	3	Popularity Award
6	Snowy	9.05	6th	4	Popularity Award
7	Fiona	8.95	7th	5	(null)
8	Coconut	8.85	8th	5	(null)
9	Dino	8.8	9th	5	(null)
10	Sunny	8.3	10th	3	(null)
11	Ricky	8.2	11th	3	(null)
12	Fluffy	7.3	12th	1	(null)
13	Cupcake	6.95	13th	2	(null)

DB_Project.sql

Worksheet | Query Builder

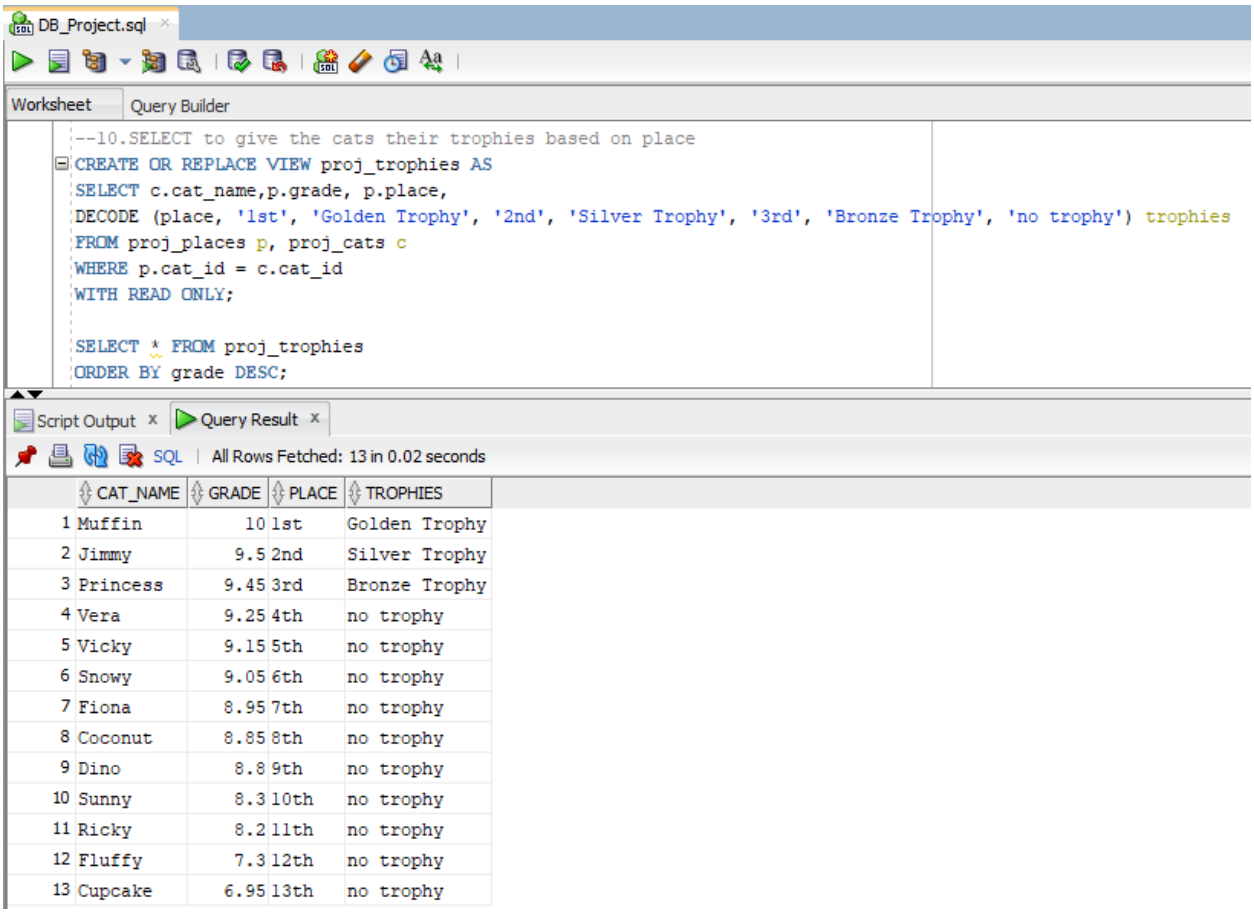
```
--9.SELECT to give the winners also a title of the best from their breed
CREATE OR REPLACE VIEW proj_best_breed AS
SELECT c.cat_name, p.grade, p.place,
CASE WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Mai%' THEN 'Best Maine Coon'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Sph%' THEN 'Best Sphynx'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Mun%' THEN 'Best Munchkin'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Bir%' THEN 'Best Birman'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Dev%' THEN 'Best Devon Rex'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Rag%' THEN 'Best Ragamuffin'
WHEN place IN (SELECT place FROM proj_places WHERE place='1st' OR place='2nd' OR place='3rd') AND breed LIKE 'Bri%' THEN 'Best British Shorthair'
ELSE NULL
END breed_awards
FROM proj_places p, proj_cats c
WHERE p.cat_id = c.cat_id
WITH READ ONLY;

SELECT * FROM proj_best_breed
ORDER BY grade DESC;
```

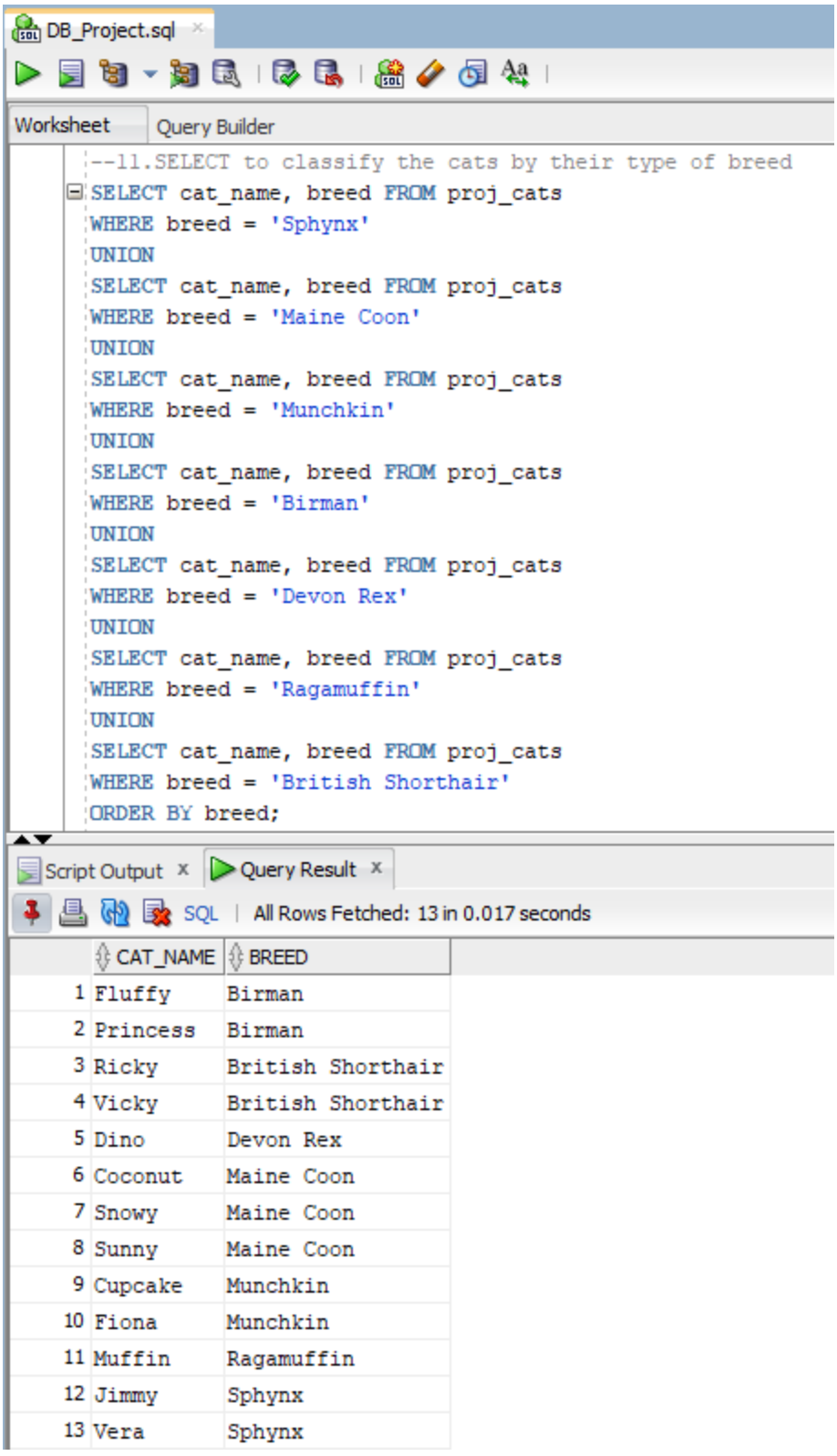
Script Output | Query Result

SQL | All Rows Fetched: 13 in 0.035 seconds

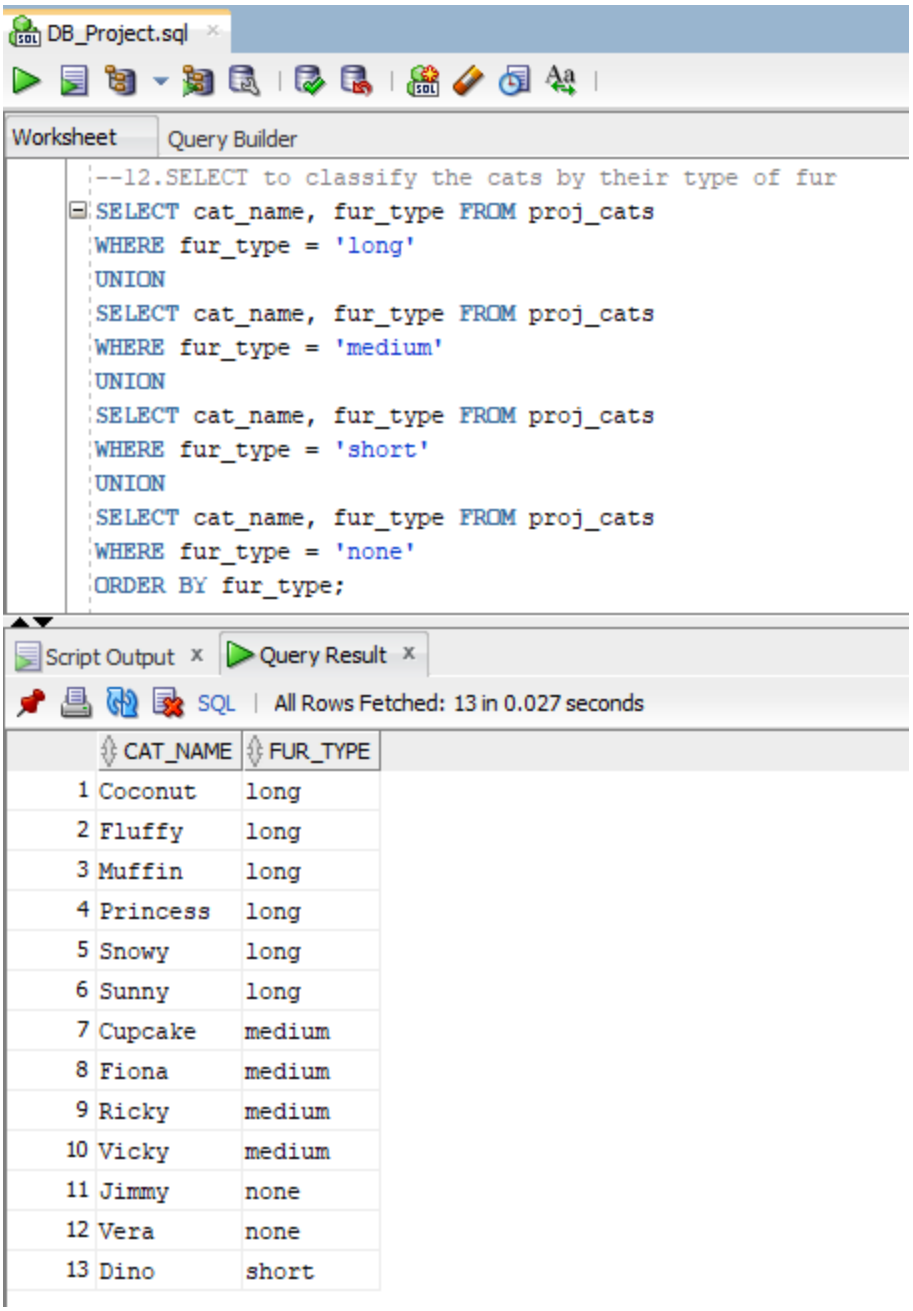
	CAT_NAME	GRADE	PLACE	BREED_AWARDS
1	Muffin	10	1st	Best Ragamuffin
2	Jimmy	9.5	2nd	Best Sphynx
3	Princess	9.45	3rd	Best Birman
4	Vera	9.25	4th	(null)
5	Vicky	9.15	5th	(null)
6	Snowy	9.05	6th	(null)
7	Fiona	8.95	7th	(null)
8	Coconut	8.85	8th	(null)
9	Dino	8.8	9th	(null)
10	Sunny	8.3	10th	(null)
11	Ricky	8.2	11th	(null)
12	Fluffy	7.3	12th	(null)
13	Cupcake	6.95	13th	(null)



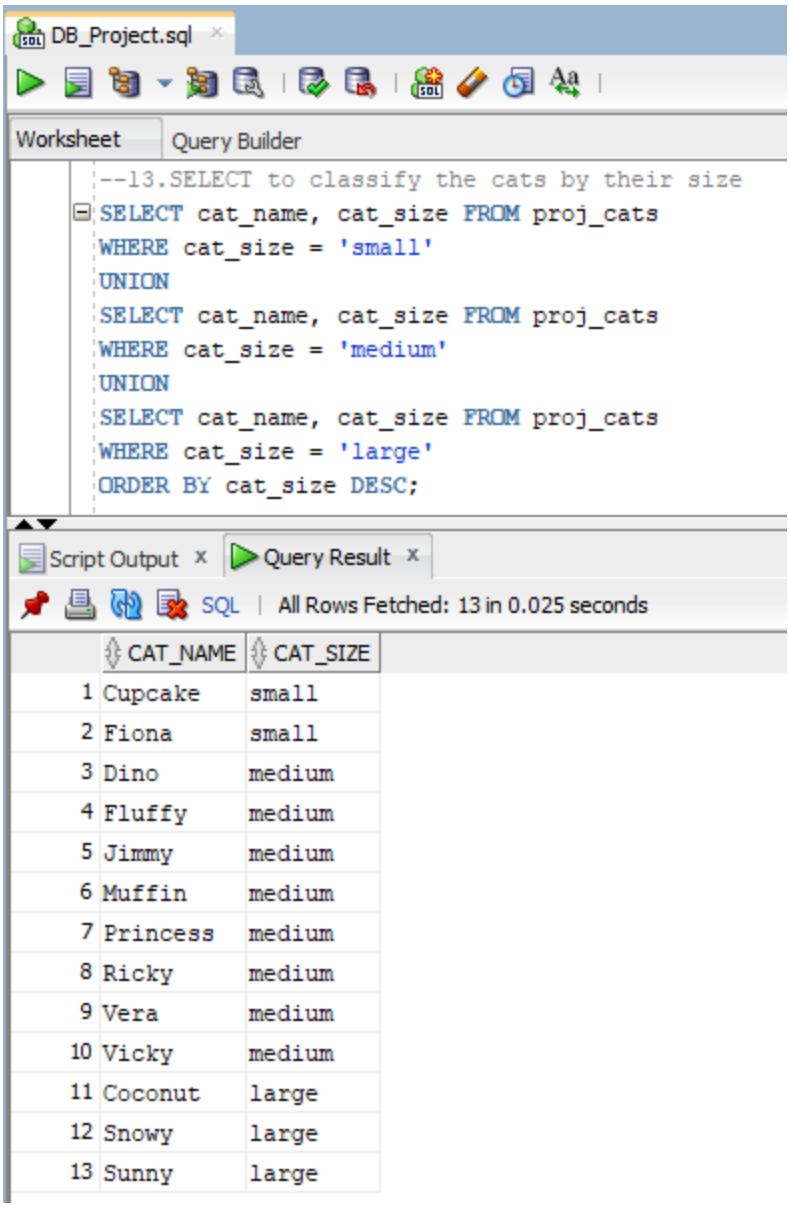
	CAT_NAME	GRADE	PLACE	TROPHIES
1	Muffin	10	1st	Golden Trophy
2	Jimmy	9.5	2nd	Silver Trophy
3	Princess	9.45	3rd	Bronze Trophy
4	Vera	9.25	4th	no trophy
5	Vicky	9.15	5th	no trophy
6	Snowy	9.05	6th	no trophy
7	Fiona	8.95	7th	no trophy
8	Coconut	8.85	8th	no trophy
9	Dino	8.8	9th	no trophy
10	Sunny	8.3	10th	no trophy
11	Ricky	8.2	11th	no trophy
12	Fluffy	7.3	12th	no trophy
13	Cupcake	6.95	13th	no trophy

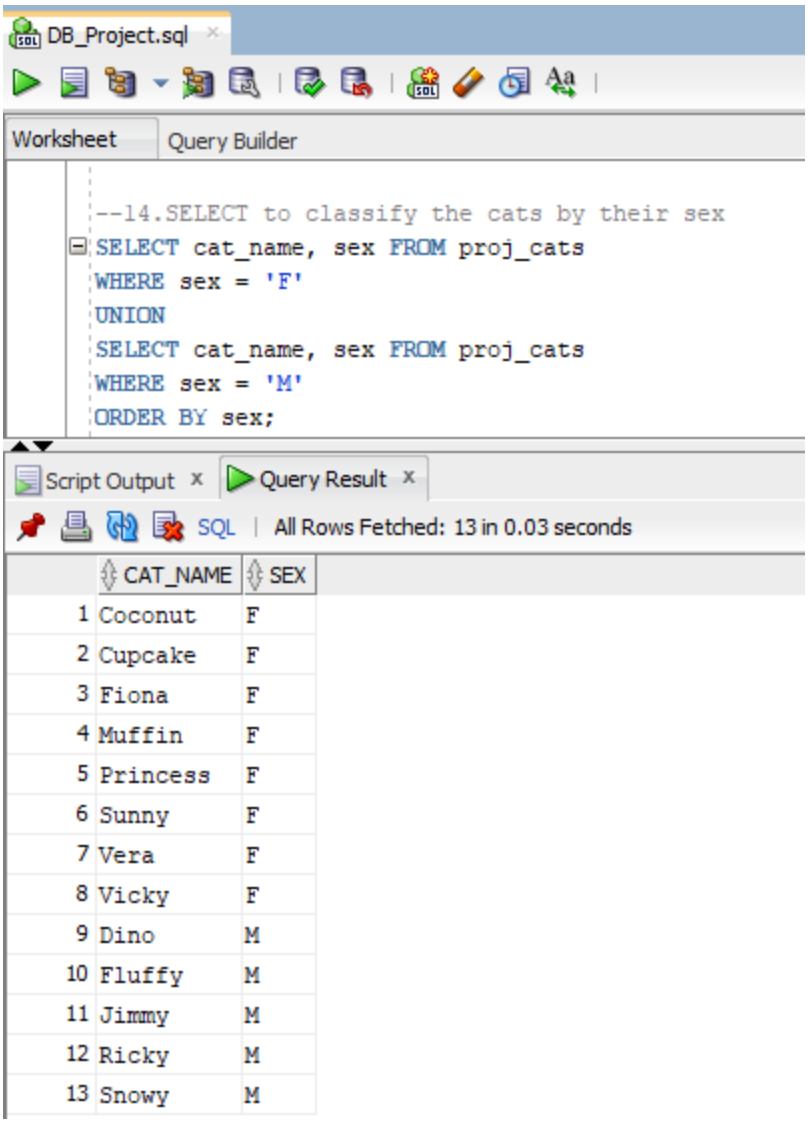


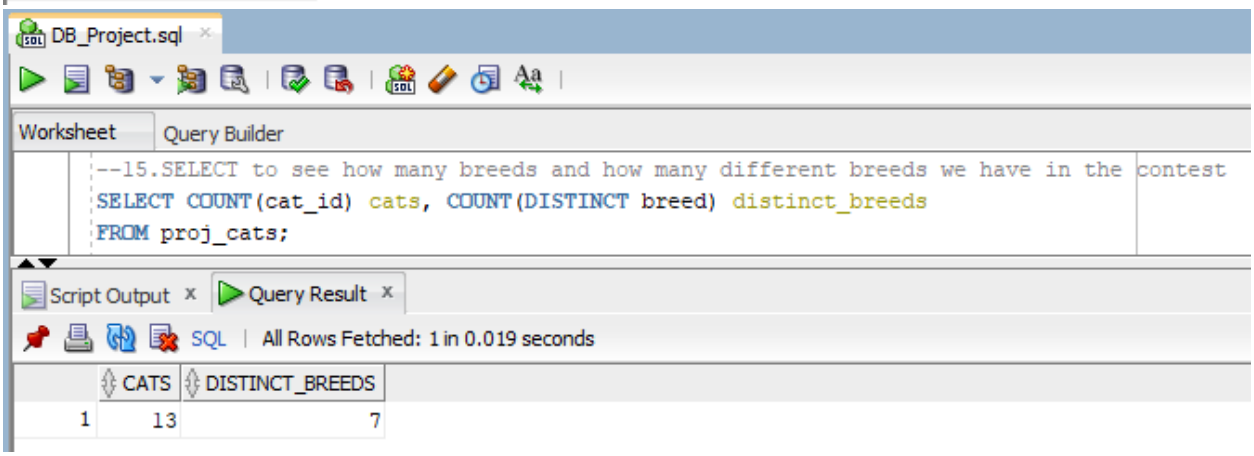
	CAT_NAME	BREED
1	Fluffy	Birman
2	Princess	Birman
3	Ricky	British Shorthair
4	Vicky	British Shorthair
5	Dino	Devon Rex
6	Coconut	Maine Coon
7	Snowy	Maine Coon
8	Sunny	Maine Coon
9	Cupcake	Munchkin
10	Fiona	Munchkin
11	Muffin	Ragamuffin
12	Jimmy	Sphynx
13	Vera	Sphynx

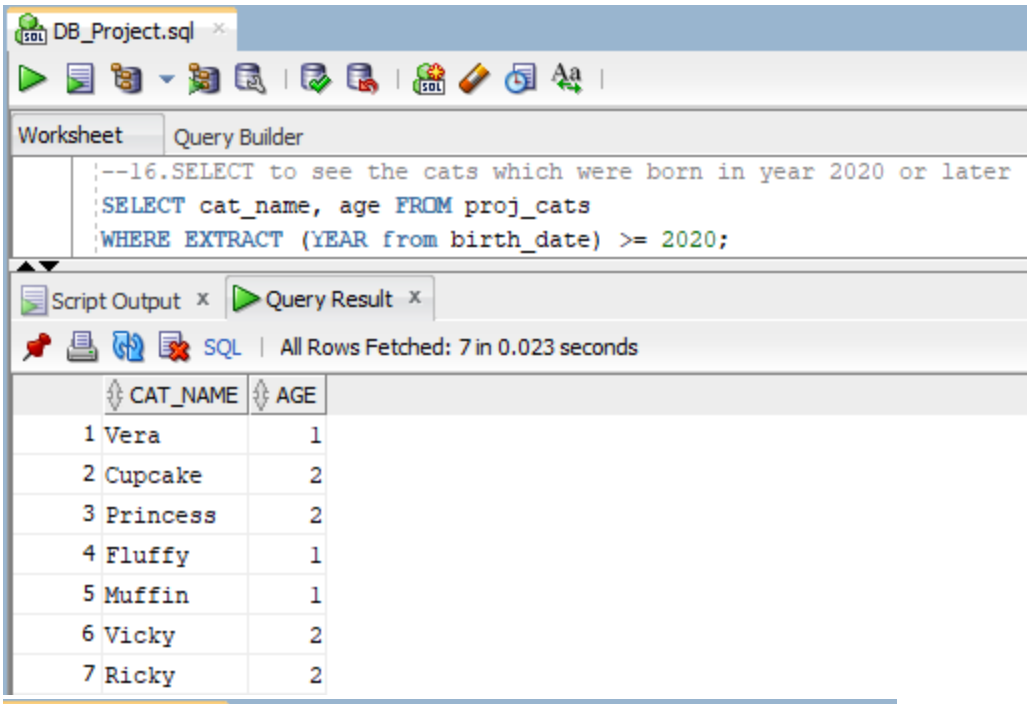


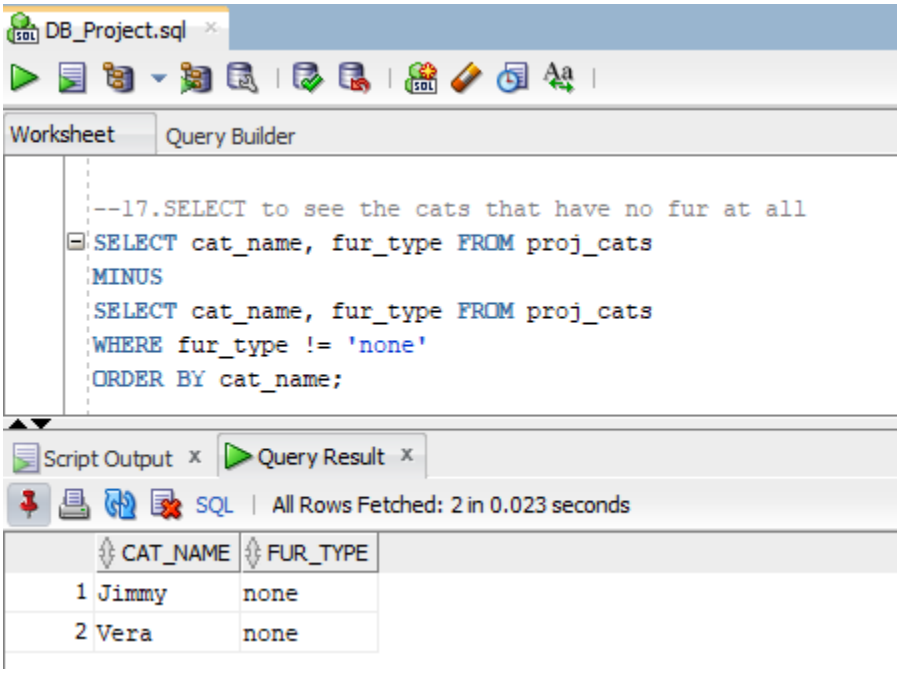
	CAT_NAME	FUR_TYPE
1	Coconut	long
2	Fluffy	long
3	Muffin	long
4	Princess	long
5	Snowy	long
6	Sunny	long
7	Cupcake	medium
8	Fiona	medium
9	Ricky	medium
10	Vicky	medium
11	Jimmy	none
12	Vera	none
13	Dino	short

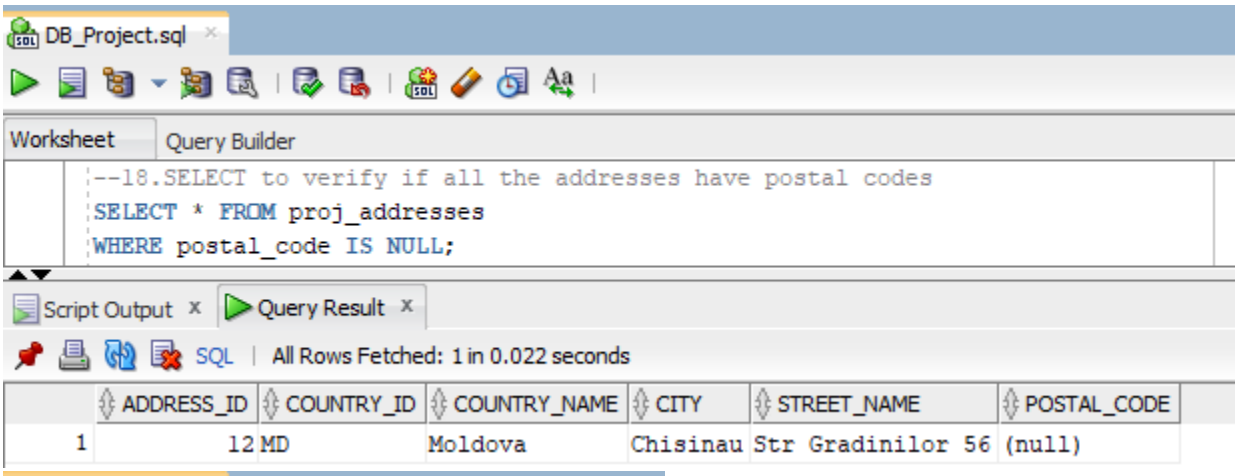

DB_Project.sql
Worksheet
Query Builder
--13.SELECT to classify the cats by their size
SELECT cat_name, cat_size FROM proj_cats
WHERE cat_size = 'small'
UNION
SELECT cat_name, cat_size FROM proj_cats
WHERE cat_size = 'medium'
UNION
SELECT cat_name, cat_size FROM proj_cats
WHERE cat_size = 'large'
ORDER BY cat_size DESC;
Script Output
Query Result
SQL
All Rows Fetched: 13 in 0.025 seconds
CAT_NAME
CAT_SIZE
1 Cupcake small
2 Fiona small
3 Dino medium
4 Fluffy medium
5 Jimmy medium
6 Muffin medium
7 Princess medium
8 Ricky medium
9 Vera medium
10 Vicky medium
11 Coconut large
12 Snowy large
13 Sunny large

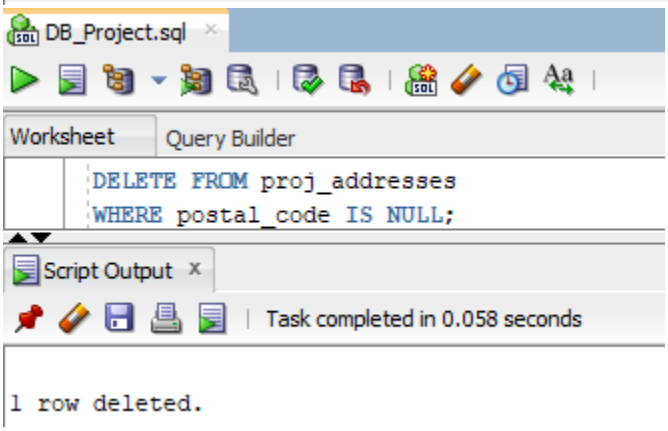

DB_Project.sql
Worksheet
Query Builder
--14.SELECT to classify the cats by their sex
SELECT cat_name, sex FROM proj_cats
WHERE sex = 'F'
UNION
SELECT cat_name, sex FROM proj_cats
WHERE sex = 'M'
ORDER BY sex;
Script Output
Query Result
SQL | All Rows Fetched: 13 in 0.03 seconds
CAT_NAME SEX
1 Coconut F
2 Cupcake F
3 Fiona F
4 Muffin F
5 Princess F
6 Sunny F
7 Vera F
8 Vicky F
9 Dino M
10 Fluffy M
11 Jimmy M
12 Ricky M
13 Snowy M

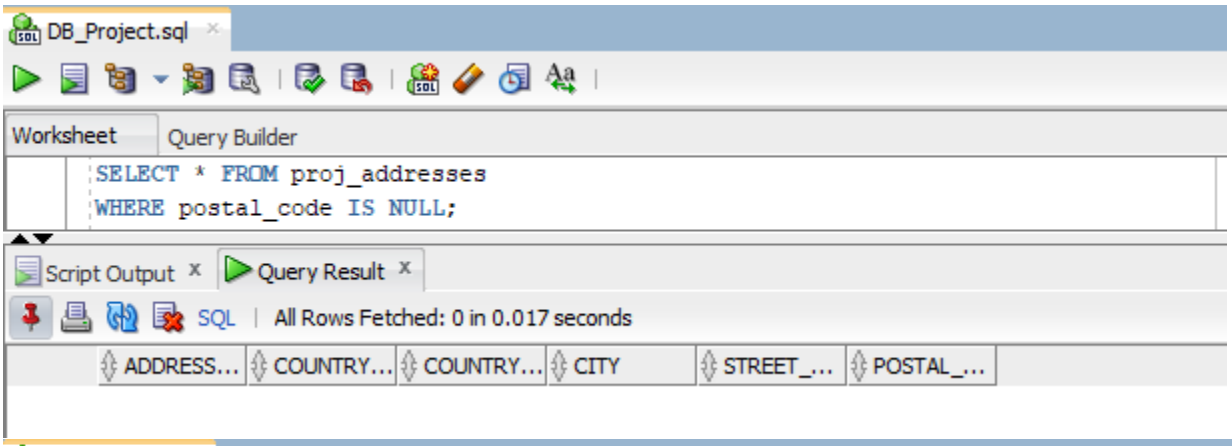

DB_Project.sql
Worksheet
Query Builder
--15.SELECT to see how many breeds and how many different breeds we have in the contest
SELECT COUNT(cat_id) cats, COUNT(DISTINCT breed) distinct_breeds
FROM proj_cats;
Script Output
Query Result
SQL | All Rows Fetched: 1 in 0.019 seconds
CATS DISTINCT_BREEDS
1 13 7

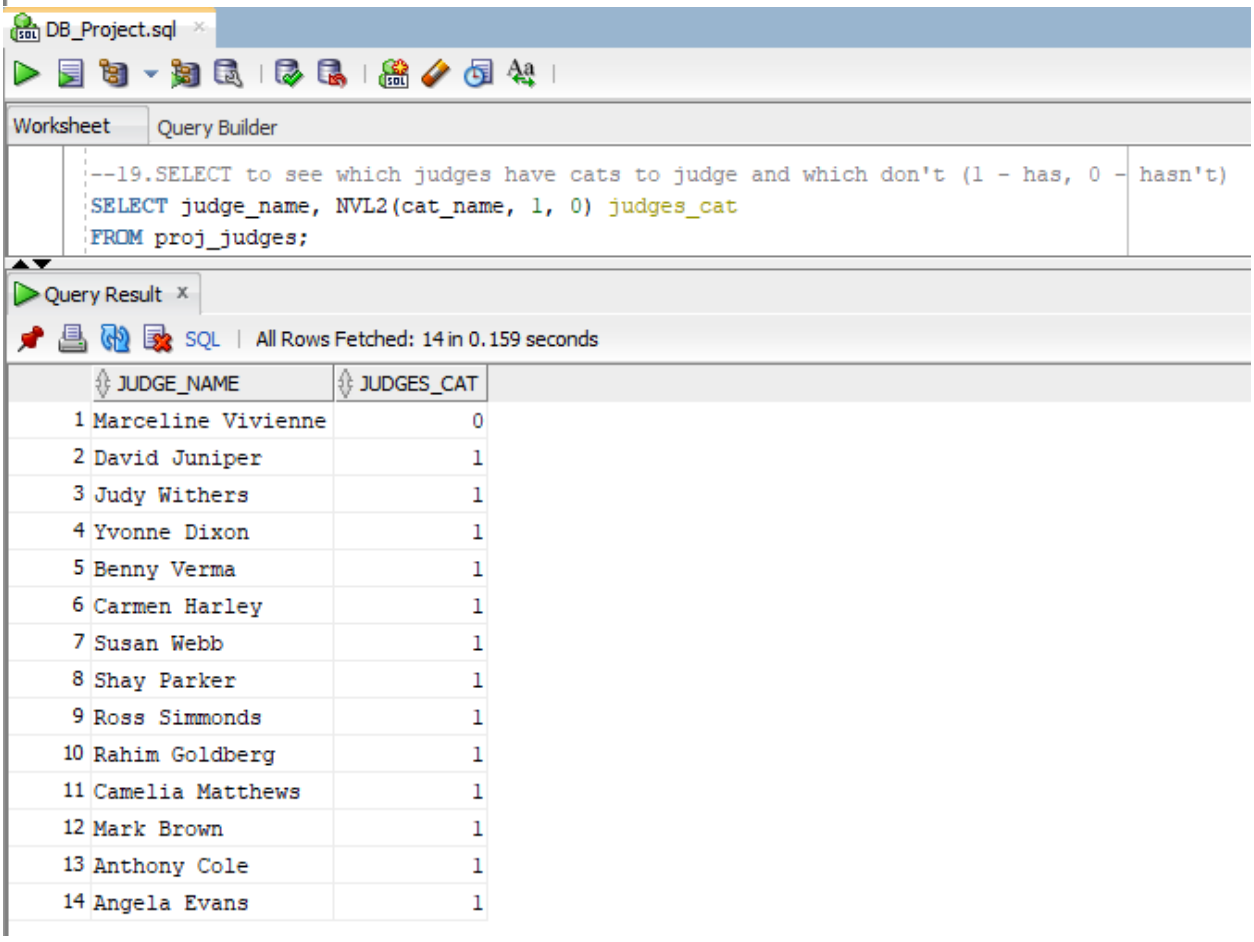

DB_Project.sql
Worksheet
Query Builder
--16.SELECT to see the cats which were born in year 2020 or later
SELECT cat_name, age FROM proj_cats
WHERE EXTRACT (YEAR from birth_date) >= 2020;
Script Output
Query Result
SQL | All Rows Fetched: 7 in 0.023 seconds
CAT_NAME AGE
1 Vera 1
2 Cupcake 2
3 Princess 2
4 Fluffy 1
5 Muffin 1
6 Vicky 2
7 Ricky 2


DB_Project.sql
Worksheet
Query Builder
--17.SELECT to see the cats that have no fur at all
SELECT cat_name, fur_type FROM proj_cats
MINUS
SELECT cat_name, fur_type FROM proj_cats
WHERE fur_type != 'none'
ORDER BY cat_name;
Script Output
Query Result
SQL | All Rows Fetched: 2 in 0.023 seconds
CAT_NAME FUR_TYPE
1 Jimmy none
2 Vera none


DB_Project.sql
Worksheet
Query Builder
--18.SELECT to verify if all the addresses have postal codes
SELECT * FROM proj_addresses
WHERE postal_code IS NULL;
Script Output
Query Result
SQL | All Rows Fetched: 1 in 0.022 seconds
ADDRESS_ID
COUNTRY_ID
COUNTRY_NAME
CITY
STREET_NAME
POSTAL_CODE
1
12
MD
Moldova
Chisinau
Str Gradinilor 56
(null)


DB_Project.sql
Worksheet
Query Builder
DELETE FROM proj_addresses
WHERE postal_code IS NULL;
Script Output
Task completed in 0.058 seconds
1 row deleted.





	JUDGE_NAME	JUDGES_CAT
1	Marceline Vivienne	0
2	David Juniper	1
3	Judy Withers	1
4	Yvonne Dixon	1
5	Benny Verma	1
6	Carmen Harley	1
7	Susan Webb	1
8	Shay Parker	1
9	Ross Simmonds	1
10	Rahim Goldberg	1
11	Camelia Matthews	1
12	Mark Brown	1
13	Anthony Cole	1
14	Angela Evans	1

DB_Project.sql

Worksheet | Query Builder

```
--20.SELECT to see the owners' initials
SELECT SUBSTR(last_name,0,1) AS "Last name initials", SUBSTR(first_name,0,1) AS "First name initials"
FROM proj_owners
ORDER BY SUBSTR(last_name,0,1);
```

Query Result

SQL | All Rows Fetched: 10 in 0.02 seconds

	Last name initials	First name initials
1	C	C
2	C	N
3	J	A
4	M	R
5	P	M
6	P	A
7	R	T
8	S	N
9	S	L
10	W	C

DB_Project.sql

Worksheet | Query Builder

```
--20.SELECT to see the owners' initials
SELECT SUBSTR(last_name,0,1) || SUBSTR(first_name,0,1) AS "Owner initials"
FROM proj_owners
ORDER BY SUBSTR(last_name,0,1);
```

Query Result

SQL | All Rows Fetched: 10 in 0.017 seconds

	Owner initials
1	CC
2	CN
3	JA
4	MR
5	PM
6	PA
7	RT
8	SN
9	SL
10	WC

DB_Project.sql

Worksheet | Query Builder

```
--21.SELECT to see if we have cats have born on the same day as the year
SELECT cat_name, birth_date, TO_CHAR(birth_date, 'DD') birth_day, TO_CHAR(birth_date, 'RRRR') birth_year FROM proj_cats
WHERE TO_CHAR(birth_date, 'DD') = TO_CHAR(birth_date, 'RR')
ORDER BY birth_date;
```

Query Result

SQL | All Rows Fetched: 0 in 0.017 seconds

CAT_NAME	BIRTH_D...	BIRTH_DAY	BIRTH_YE...

DB_Project.sql

Worksheet | Query Builder

```
--22.SELECT to see what places got the juniors (0-2 years)
SELECT c.cat_name, c.birth_date, c.age, p.place
FROM proj_cats c, proj_places p
WHERE c.cat_id = p.cat_id
AND birth_date BETWEEN TO_DATE('01-01-2020','DD-MM-RRRR') AND TO_DATE('31-12-2022','DD-MM-RRRR')
ORDER BY grade DESC;
```

Query Result

SQL | All Rows Fetched: 7 in 0.019 seconds

	CAT_NAME	BIRTH_DATE	AGE	PLACE
1	Muffin	07-DEC-21	1	1st
2	Princess	28-FEB-20	2	3rd
3	Vera	14-MAY-21	1	4th
4	Vicky	04-JAN-20	2	5th
5	Ricky	04-JAN-20	2	11th
6	Fluffy	27-JUL-21	1	12th
7	Cupcake	15-DEC-20	2	13th

DB_Project.sql

Worksheet | Query Builder

```
--23.Cats younger than 1, 2 OR 3 years old
SELECT cat_name, age
FROM   proj_cats
WHERE  age < ANY (1, 2, 3)
ORDER BY age;
```

Query Result

SQL | All Rows Fetched: 7 in 0.023 seconds

	CAT_NAME	AGE
1	Vera	1
2	Fluffy	1
3	Muffin	1
4	Ricky	2
5	Vicky	2
6	Princess	2
7	Cupcake	2

DB_Project.sql

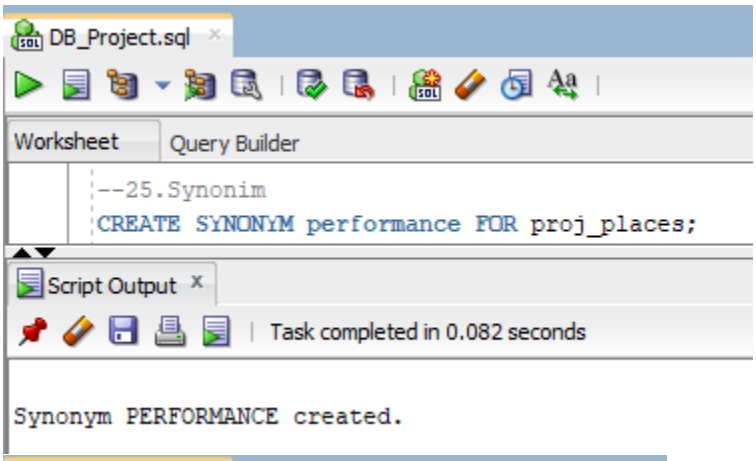
Worksheet | Query Builder

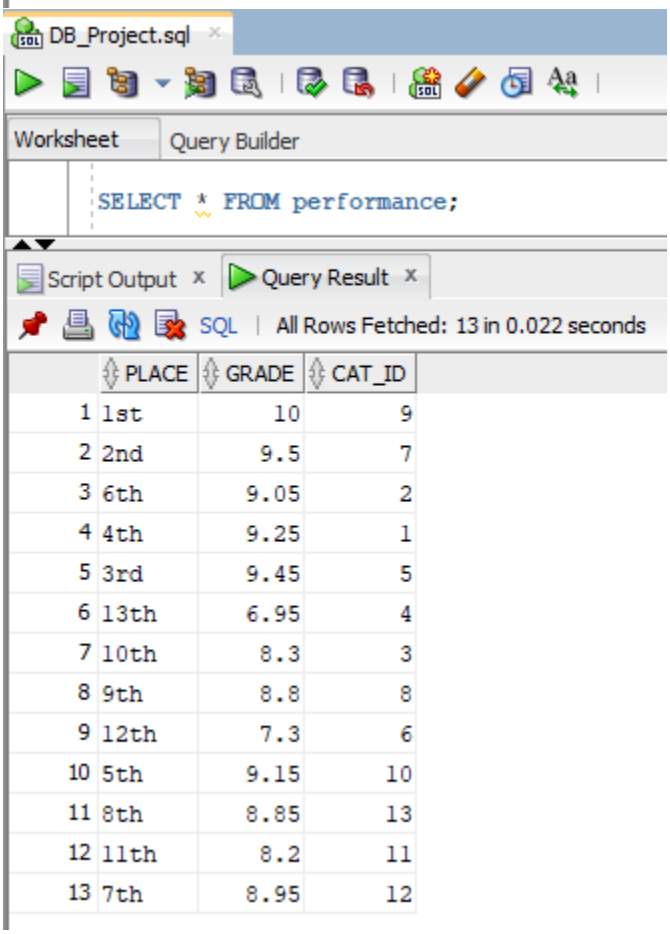
```
--24.Cats older than 1, 2 AND 3 years old
SELECT cat_name, age
FROM   proj_cats
WHERE  age > ALL (1, 2, 3)
ORDER BY age;
```

Query Result

SQL | All Rows Fetched: 4 in 0.023 seconds

	CAT_NAME	AGE
1	Snowy	4
2	Fiona	5
3	Coconut	5
4	Dino	5





	PLACE	GRADE	CAT_ID
1	1st	10	9
2	2nd	9.5	7
3	6th	9.05	2
4	4th	9.25	1
5	3rd	9.45	5
6	13th	6.95	4
7	10th	8.3	3
8	9th	8.8	8
9	12th	7.3	6
10	5th	9.15	10
11	8th	8.85	13
12	11th	8.2	11
13	7th	8.95	12

DB_Project.sql

Worksheet | Query Builder

```
--(Hierarchical query 1)
--26.Sorting the judges by hierarchical level:
SELECT judge_id "Judge ID", judge_name "Judge name", manager_id "Manager ID", LEVEL "Level"
FROM proj_judges
START WITH judge_id = 1
CONNECT BY nocycle PRIOR judge_id = manager_id
ORDER BY LEVEL;
```

Query Result

SQL | All Rows Fetched: 14 in 0.02 seconds

	Judge ID	Judge name	Manager ID	Level
1	1	Marceline Vivienne	(null)	1
2	2	David Juniper	1	2
3	14	Angela Evans	1	2
4	3	Judy Withers	2	3
5	4	Yvonne Dixon	2	3
6	5	Benny Verma	3	4
7	6	Carmen Harley	4	4
8	8	Shay Parker	6	5
9	7	Susan Webb	5	5
10	10	Rahim Goldberg	8	6
11	9	Ross Simmonds	7	6
12	11	Camelia Matthews	9	7
13	12	Mark Brown	10	7
14	13	Anthony Cole	11	8

DB_Project.sql

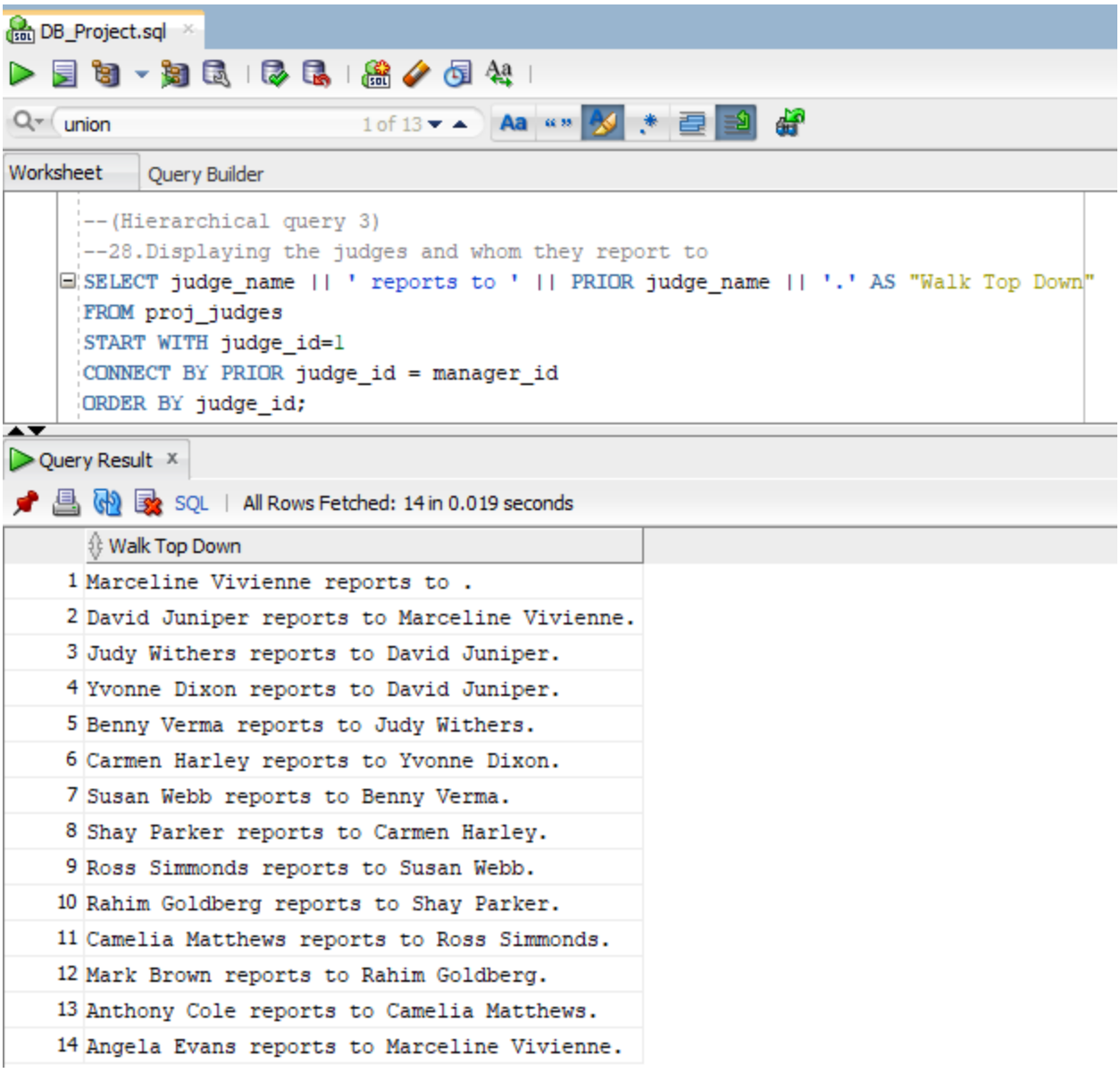
Worksheet | Query Builder

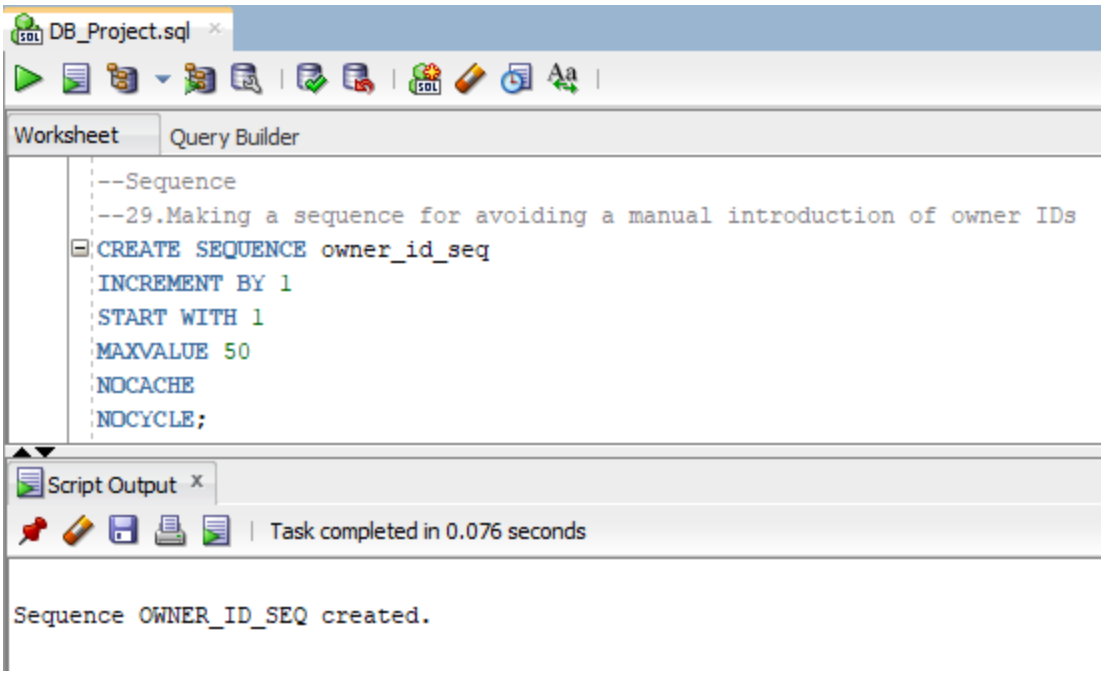
```
--(Hierarchical query 2)
--27.Displaying the judges subordinated to the root record, specifying for each judge the manager, the number of superiors and all his superiors
SELECT judge_id "Judge ID", judge_name "Judge name", LEVEL-1 "No of superiors", SYS_CONNECT_BY_PATH(judge_name, '--') "Superiors"
FROM proj_judges
CONNECT BY nocycle PRIOR judge_id = manager_id
START WITH judge_id=1
ORDER BY LEVEL-1;
```

Query Result

SQL | All Rows Fetched: 14 in 0.019 seconds

	Judge ID	Judge name	No of superiors	Superiors
1	1	Marceline Vivienne	0	--Marceline Vivienne
2	2	David Juniper	1	--Marceline Vivienne--David Juniper
3	14	Angela Evans	1	--Marceline Vivienne--Angela Evans
4	3	Judy Withers	2	--Marceline Vivienne--David Juniper--Judy Withers
5	4	Yvonne Dixon	2	--Marceline Vivienne--David Juniper--Yvonne Dixon
6	5	Benny Verma	3	--Marceline Vivienne--David Juniper--Judy Withers--Benny Verma
7	6	Carmen Harley	3	--Marceline Vivienne--David Juniper--Yvonne Dixon--Carmen Harley
8	8	Shay Parker	4	--Marceline Vivienne--David Juniper--Yvonne Dixon--Carmen Harley--Shay Parker
9	7	Susan Webb	4	--Marceline Vivienne--David Juniper--Judy Withers--Benny Verma--Susan Webb
10	10	Rahim Goldberg	5	--Marceline Vivienne--David Juniper--Yvonne Dixon--Carmen Harley--Shay Parker--Rahim Goldberg
11	9	Ross Simmonds	5	--Marceline Vivienne--David Juniper--Judy Withers--Benny Verma--Susan Webb--Ross Simmonds
12	11	Camelia Matthews	6	--Marceline Vivienne--David Juniper--Judy Withers--Benny Verma--Susan Webb--Ross Simmonds--Camelia Matthews
13	12	Mark Brown	6	--Marceline Vivienne--David Juniper--Yvonne Dixon--Carmen Harley--Shay Parker--Rahim Goldberg--Mark Brown
14	13	Anthony Cole	7	--Marceline Vivienne--David Juniper--Judy Withers--Benny Verma--Susan Webb--Ross Simmonds--Camelia Matthews--Anthony Cole


DB_Project.sql
union
1 of 13
Worksheet
Query Builder
--(Hierarchical query 3)
--28.Displaying the judges and whom they report to
SELECT judge_name || ' reports to ' || PRIOR judge_name || '.' AS "Walk Top Down"
FROM proj_judges
START WITH judge_id=1
CONNECT BY PRIOR judge_id = manager_id
ORDER BY judge_id;
Query Result
SQL
All Rows Fetched: 14 in 0.019 seconds
Walk Top Down
1 Marceline Vivienne reports to .
2 David Juniper reports to Marceline Vivienne.
3 Judy Withers reports to David Juniper.
4 Yvonne Dixon reports to David Juniper.
5 Benny Verma reports to Judy Withers.
6 Carmen Harley reports to Yvonne Dixon.
7 Susan Webb reports to Benny Verma.
8 Shay Parker reports to Carmen Harley.
9 Ross Simmonds reports to Susan Webb.
10 Rahim Goldberg reports to Shay Parker.
11 Camelia Matthews reports to Ross Simmonds.
12 Mark Brown reports to Rahim Goldberg.
13 Anthony Cole reports to Camelia Matthews.
14 Angela Evans reports to Marceline Vivienne.


DB_Project.sql
Worksheet
Query Builder
--Sequence
--29.Making a sequence for avoiding a manual introduction of owner IDs
CREATE SEQUENCE owner_id_seq
INCREMENT BY 1
START WITH 1
MAXVALUE 50
NOCACHE
NOCYCLE;
Script Output
Task completed in 0.076 seconds
Sequence OWNER_ID_SEQ created.