Australian Open Database

1. Topic Definition

This database will contain the results of all men's and women's singles finals of the Australian Open tennis tournament, the prizes expressed in Australian dollars for the top two positions, and some data about the players who participated in the finals, from 2001 to 2020.

The finals will be listed in a single table for both men and women, including the year of the tournament, category (men or women), the names of the two competing tennis players, and the match result with set details. The match result interpretation follows these rules: instead of the usual two winning sets, three winning sets are required to win a game. Each set must include six games, and each game requires winning four points, each worth 15 points, totaling 60 points, with the opponent having at most 30 points (i.e., winning a maximum of two rallies).

If the score is 45-45, the loser of the next rally loses 15 points, resetting their score to 30, while the other player retains 45 points. If the player with 45 points wins, they take the game; otherwise, it returns to 45-45, and the process repeats. If both players have won five games (5-5), the set cannot be won with six games, and two possibilities arise: either one player wins two consecutive games to achieve a 7-5 victory or both players win one game each, resulting in a 6-6 tie. In this case, a "Tie-Break" follows, where each rally is worth one point, and a minimum of seven points with a two-point difference is required to win (e.g., 8-6 or 13-11).

If both players win two sets each, the fifth decisive set does not include a tie-break; the player must win by a minimum of six games and a two-game difference. If the result includes "retired," it means the losing player withdrew for some reason.

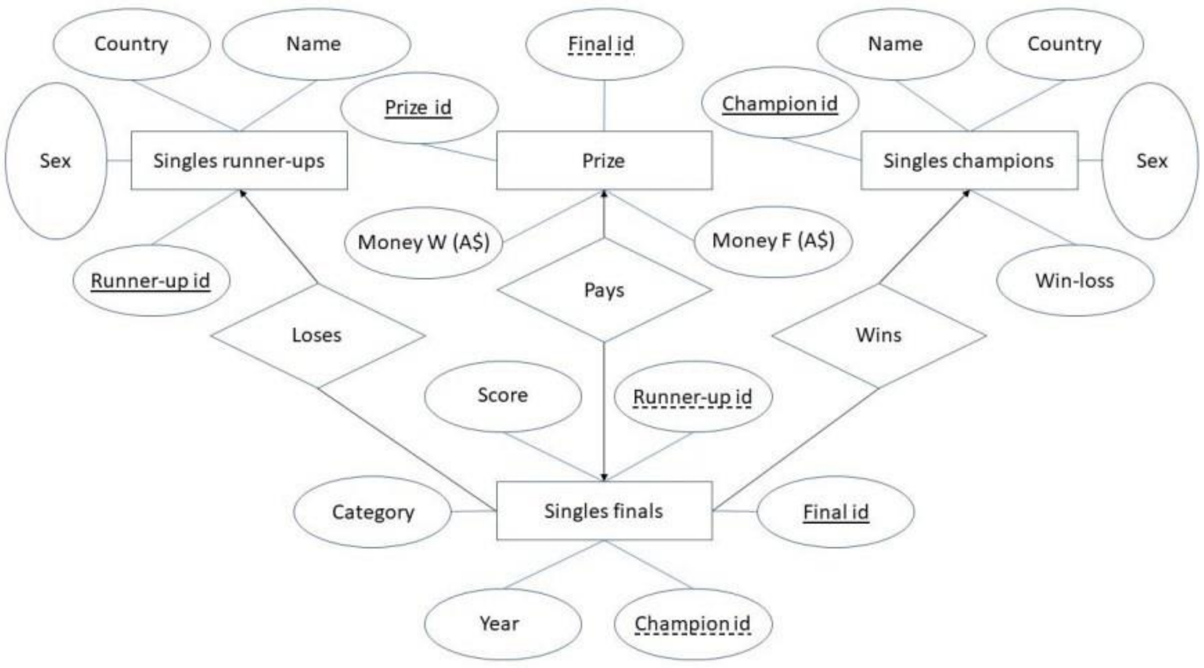
In addition, a separate table will list all male and female winners, showing the players' names, nationalities, category, and the number of finals played and won. A similar table will be created for all runners-up, except they won't have "played-won finals" fields.

A fourth table will store the prizes (1st and 2nd place) in Australian dollars, broken down by year, category, and position.
The database will consist of these four tables.

IDs in the tables will follow specific rules:

- Final ID: A three-digit number where the first digit is 1 for men's finals and 2 for women's finals, followed by the last two digits of the year (e.g., 216 for the 2016 women's final).
- Runner-up/Champion ID: The first two initials of the player's first and last name in uppercase, separated by an underscore (e.g., Rafael Nadal -> RA_NA). In case of duplicate names/IDs, the player's country code or another unique identifier (maximum three characters) will be appended.

2. E/R Model



3. Tables and Relationships

Tables and Relationships:

- **Singles Finals** *(Final ID, Year, Champion ID, Runner-up ID, Category, Score)*
    - **Primary Key**: Final ID
    - **Foreign Keys**:
        - Champion ID refers to the primary key in the *Singles Champions* table.
        - Runner-up ID refers to the primary key in the *Singles Runner-ups* table.
- **Singles Champions** *(Champion ID, Name, Country, Win-Loss, Sex)*
    - **Primary Key**: Champion ID
- **Singles Runner-ups** *(Runner-up ID, Name, Country, Sex)*
    - **Primary Key**: Runner-up ID
- **Prize** *(Prize ID, Final ID, Money W (AUD), Money F (AUD))*
    - **Primary Key**: Prize ID
    - **Foreign Key**: Final ID references Final ID in *Singles Finals*.

4. Checking and Normalizing Relations

- The fields in all relations are atomic, so they satisfy 1NF.
- Each relation is in 1NF and has no partial dependencies, thus meeting 2NF.
- Relations in 2NF do not have transitive dependencies among non-primary attributes, satisfying 3NF.

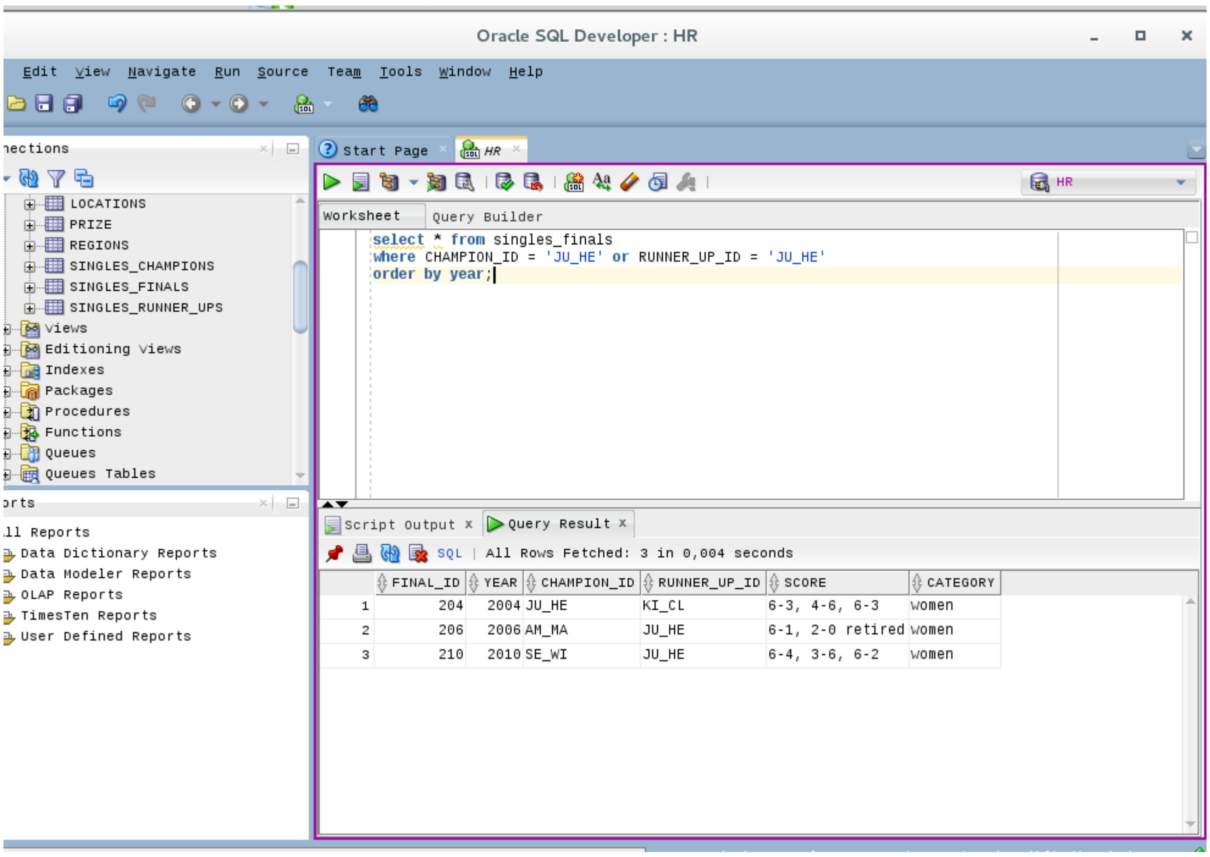
5. Table Schemas

- **Singles Runner-ups (26 Records)**
    - **Runner_up_ID**: `varchar2(9)` (Primary Key)
    - **Name**: `varchar2(100)`
    - **Country**: `varchar2(100)`
    - **Sex**: `char(1)` (Values: 'M' or 'F')
- **Singles Champions (14 Records)**
    - **Champion_ID**: `varchar2(9)` (Primary Key)
    - **Name**: `varchar2(100)`
    - **Country**: `varchar2(100)`

    - **Win_Loss**: `varchar2(7)`
    - **Sex**: `char(1)` (Values: 'M' or 'F')
- **Singles Finals (40 Records)**
    - **Final_ID**: `number(3)`
    - **Year**: `number(4)`
    - **Champion_ID**: `varchar2(9)` (Foreign Key)
    - **Runner_up_ID**: `varchar2(9)` (Foreign Key)
    - **Score**: `varchar2(100)`
    - **Category**: `varchar2(5)` (Values: 'MEN' or 'WOMEN')
- **Prize (22 Records)**
    - **Prize_ID**: `number(3)` (Primary Key)
    - **Final_ID**: `number(3)` (Foreign Key)
    - **Money_W_AUD**: `number(8)`
    - **Money_F_AUD**: `number(8)`

6. Simple and Multi-Table Queries

- **List all finals involving Justine Henin**, including all related data, in chronological order.

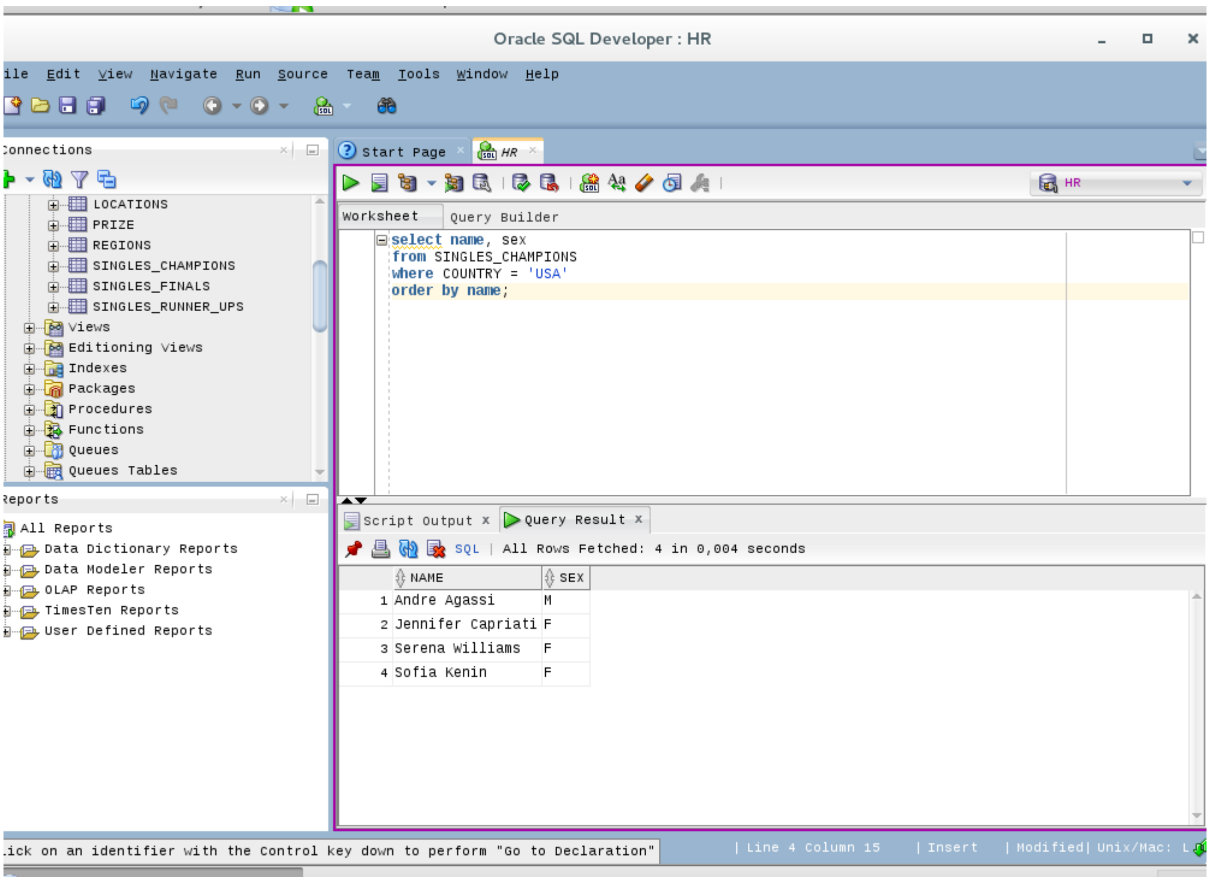


```
select * from singles_finals
where CHAMPION_ID = 'JU_HE' or RUNNER_UP_ID = 'JU_HE'
order by year;
```

All Rows Fetched: 3 in 0,004 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	SCORE	CATEGORY
1	204	2004	JU_HE	KI_CL	6-3, 4-6, 6-3	women
2	206	2006	AM_MA	JU_HE	6-1, 2-0 retired	women
3	210	2010	SE_WI	JU_HE	6-4, 3-6, 6-2	women

- **Display the names and genders of all American players**, sorted alphabetically.

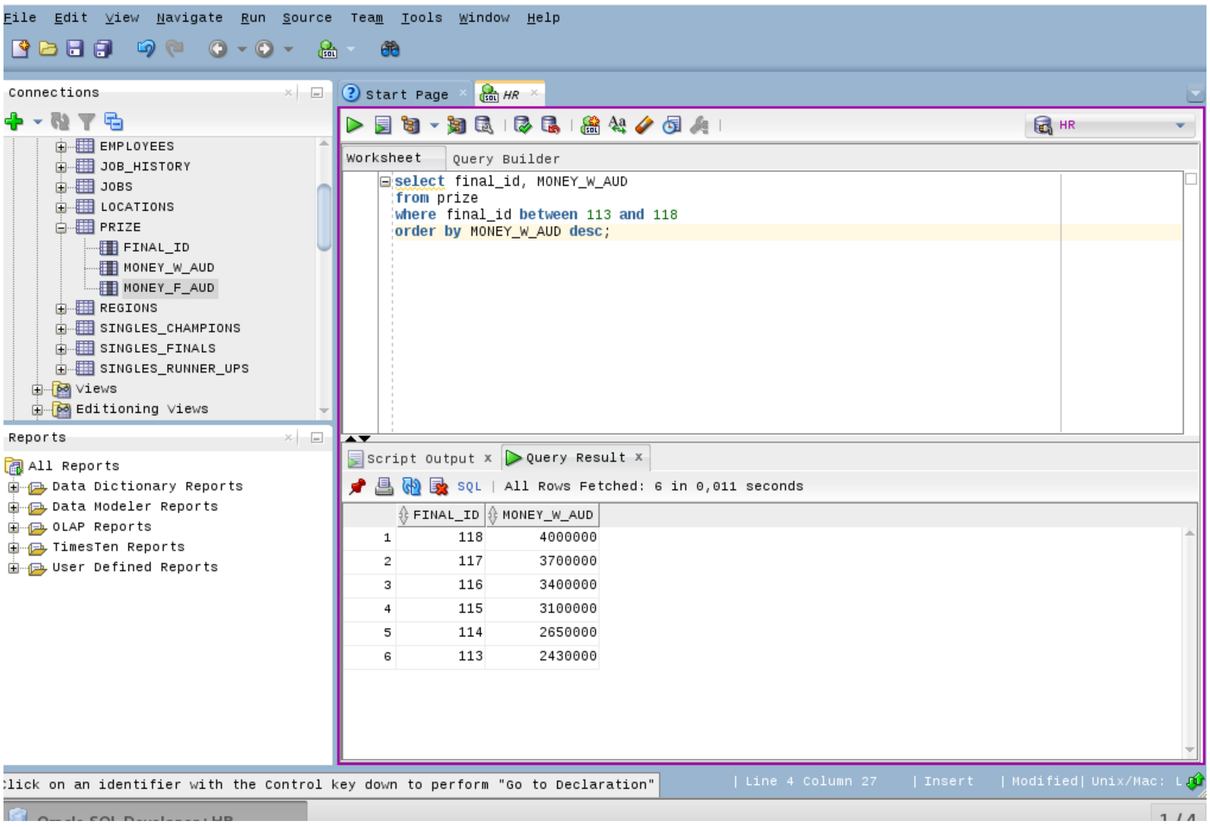


```
select name, sex
from SINGLES_CHAMPIONS
where COUNTRY = 'USA'
order by name;
```

All Rows Fetched: 4 in 0,004 seconds

```
   NAME              SEX
 1 Andre Agassi      M
 2 Jennifer Capriati F
 3 Serena Williams   F
 4 Sofia Kenin       F
```

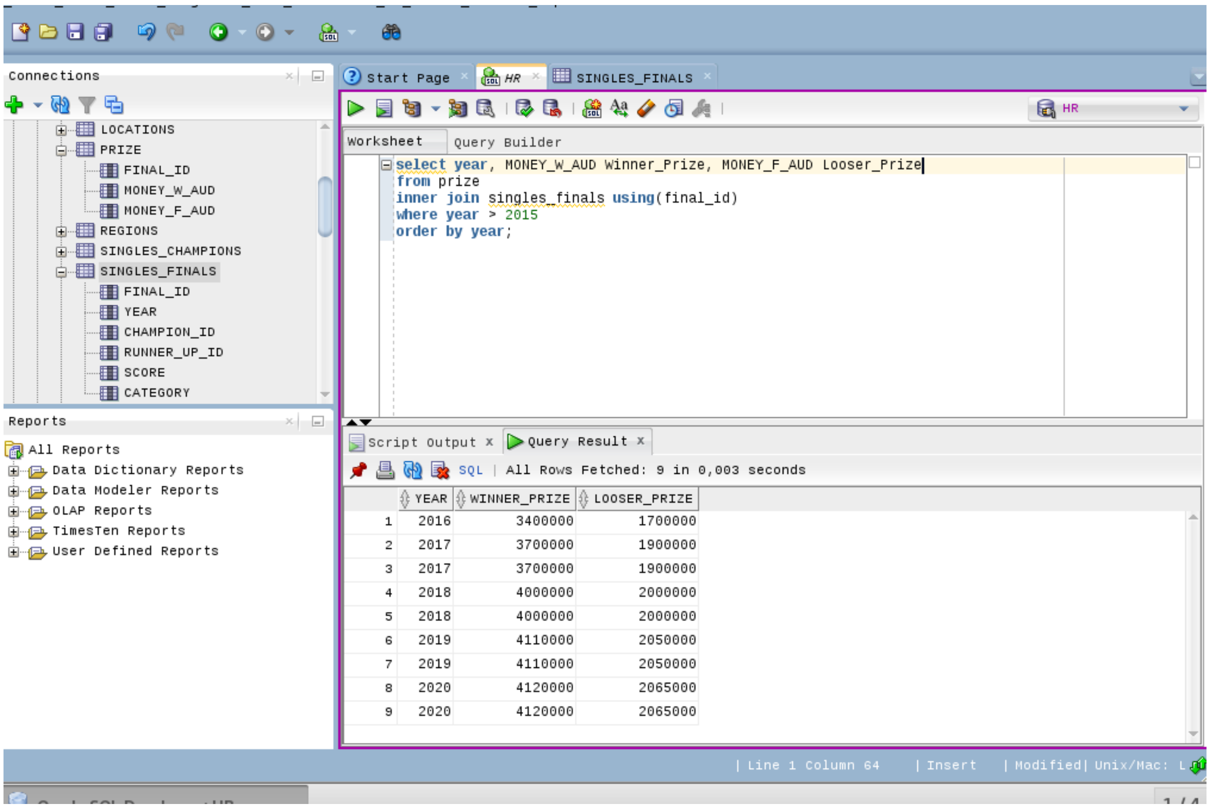
- **List the final IDs and winners' prizes** for finals with IDs between 113 and 118, sorted by prize in descending order.



```
select final_id, MONEY_W_AUD
from prize
where final_id between 113 and 118
order by MONEY_W_AUD desc;
```

```
All Rows Fetched: 6 in 0,011 seconds

    FINAL_ID  MONEY_W_AUD
1        118      4000000
2        117      3700000
3        116      3400000
4        115      3100000
5        114      2650000
6        113      2430000
```

- **Show the winners' and runners-up's prizes**, sorted by year.

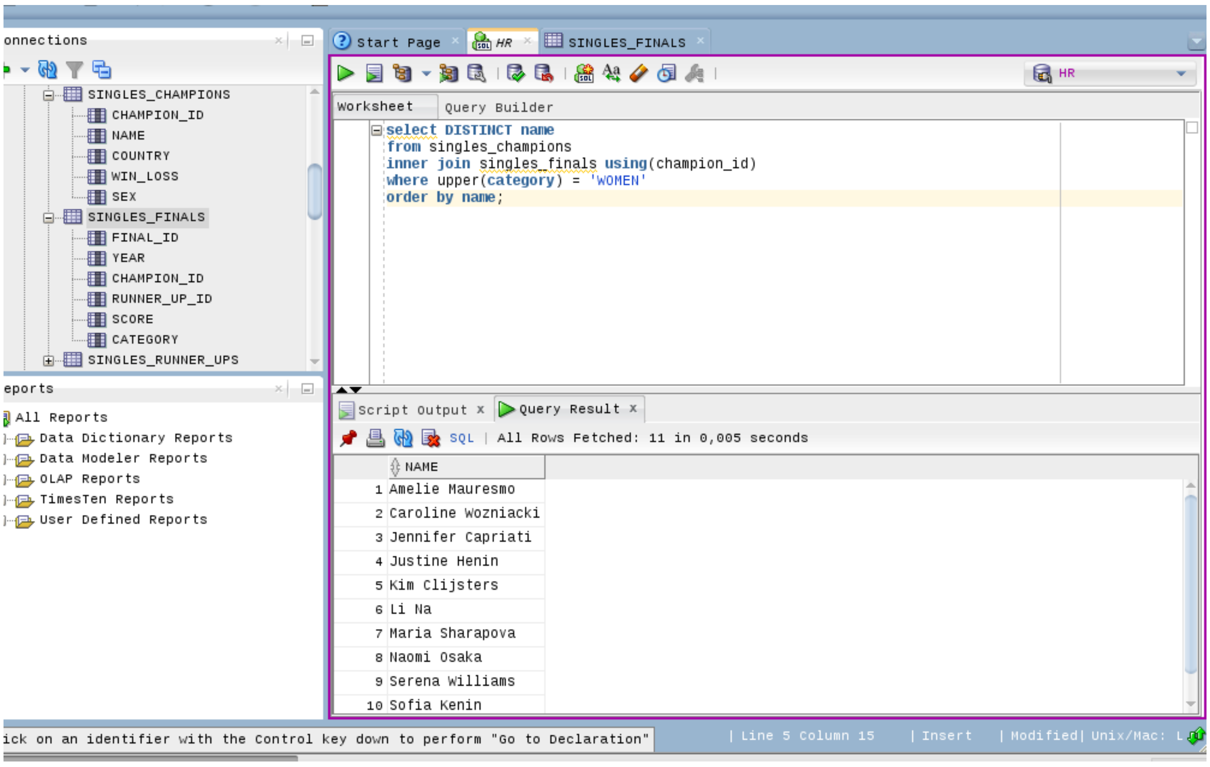


```
select year, MONEY_W_AUD Winner_Prize, MONEY_F_AUD Looser_Prize
from prize
inner join singles_finals using(final_id)
where year > 2015
order by year;
```

All Rows Fetched: 9 in 0,003 seconds

```
    YEAR  WINNER_PRIZE  LOOSER_PRIZE
1   2016       3400000       1700000
2   2017       3700000       1900000
3   2017       3700000       1900000
4   2018       4000000       2000000
5   2018       4000000       2000000
6   2019       4110000       2050000
7   2019       4110000       2050000
8   2020       4120000       2065000
9   2020       4120000       2065000
```

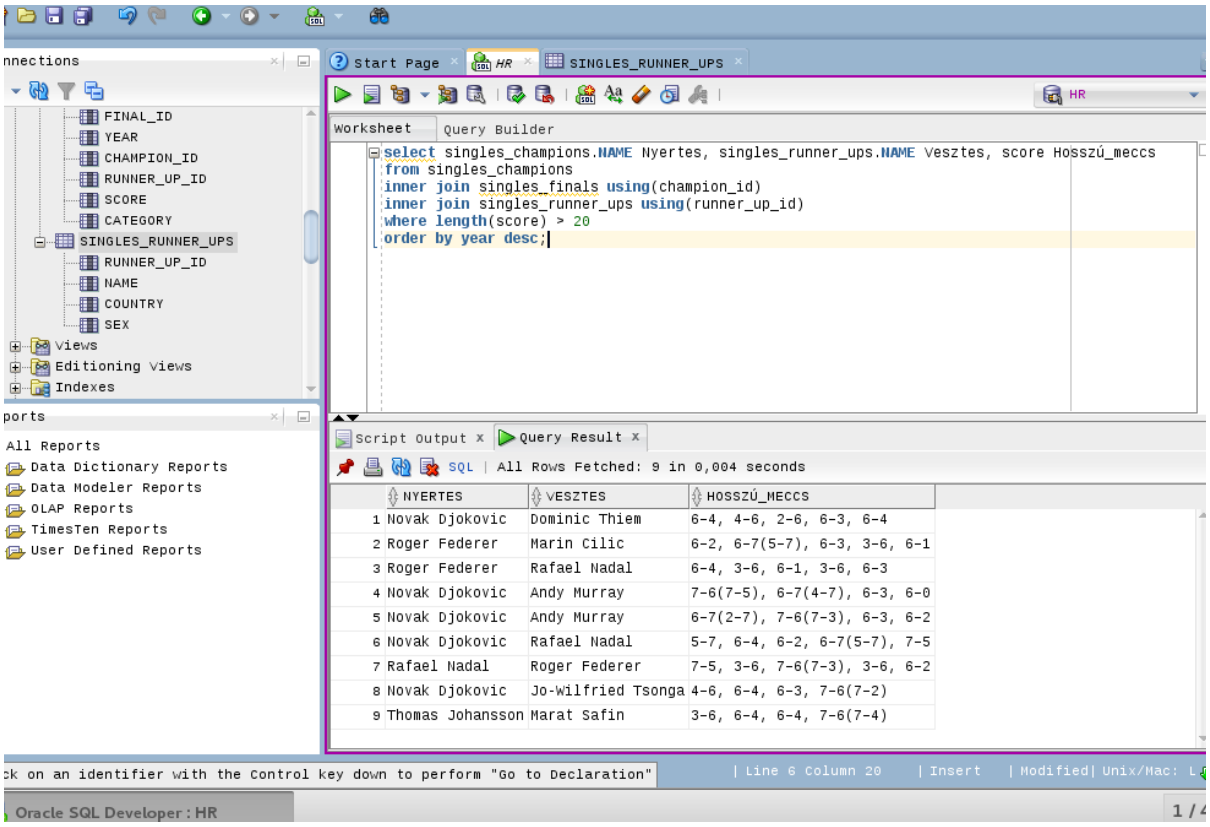
- **List all female players who have won the tournament**, ensuring each name appears only once.



```
select DISTINCT name
from singles_champions
inner join singles_finals using(champion_id)
where upper(category) = 'WOMEN'
order by name;
```

```
All Rows Fetched: 11 in 0,005 seconds

   NAME
1  Amelie Mauresmo
2  Caroline Wozniacki
3  Jennifer Capriati
4  Justine Henin
5  Kim Clijsters
6  Li Na
7  Maria Sharapova
8  Naomi Osaka
9  Serena Williams
10 Sofia Kenin
```

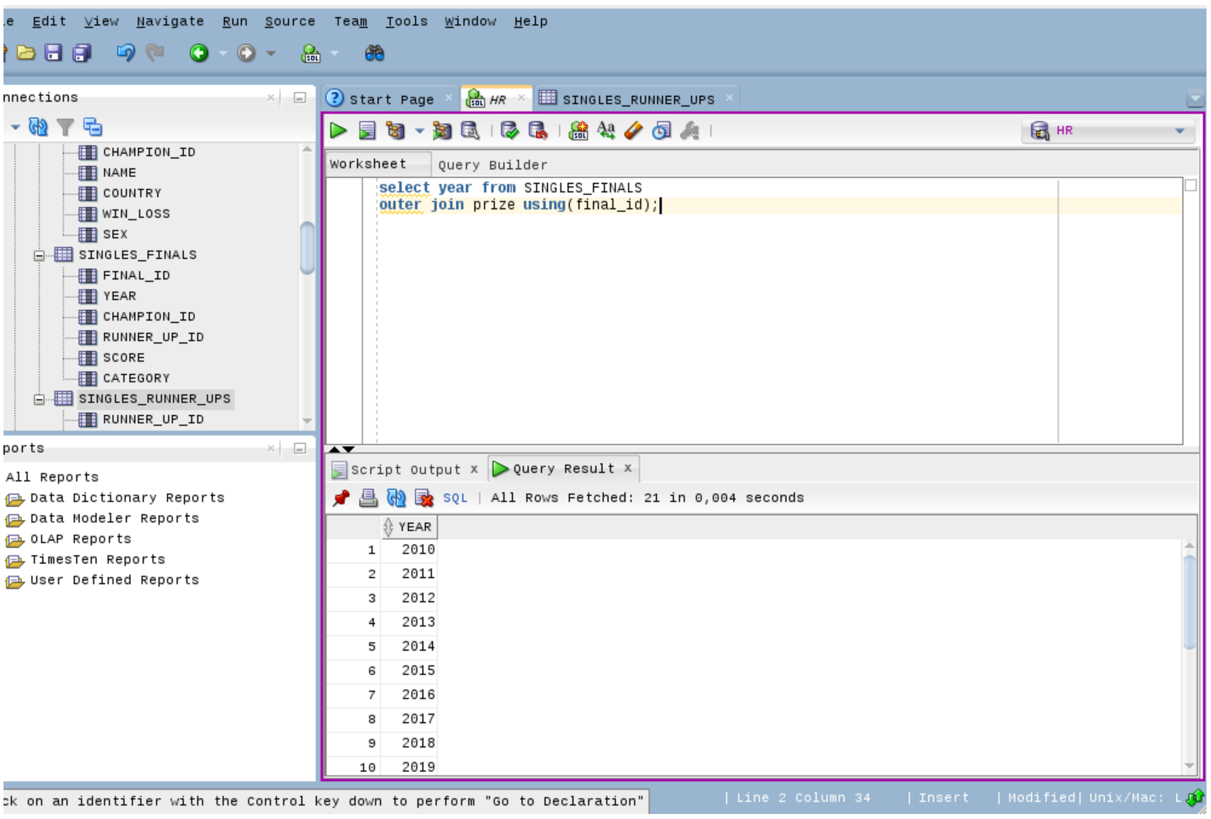
- **Identify the longest finals** based on the number of sets, displaying the names of winners and runners-up, sorted in reverse chronological order.



```sql
select singles_champions.NAME Nyertes, singles_runner_ups.NAME Vesztes, score Hosszú_meccs
from singles_champions
inner join singles_finals using(champion_id)
inner join singles_runner_ups using(runner_up_id)
where length(score) > 20
order by year desc;
```

```
All Rows Fetched: 9 in 0,004 seconds

   NYERTES          VESZTES             HOSSZÚ_MECCS
 1 Novak Djokovic   Dominic Thiem       6-4, 4-6, 2-6, 6-3, 6-4
 2 Roger Federer    Marin Cilic         6-2, 6-7(5-7), 6-3, 3-6, 6-1
 3 Roger Federer    Rafael Nadal        6-4, 3-6, 6-1, 3-6, 6-3
 4 Novak Djokovic   Andy Murray         7-6(7-5), 6-7(4-7), 6-3, 6-0
 5 Novak Djokovic   Andy Murray         6-7(2-7), 7-6(7-3), 6-3, 6-2
 6 Novak Djokovic   Rafael Nadal        5-7, 6-4, 6-2, 6-7(5-7), 7-5
 7 Rafael Nadal     Roger Federer       7-5, 3-6, 7-6(7-3), 3-6, 6-2
 8 Novak Djokovic   Jo-Wilfried Tsonga  4-6, 6-4, 6-3, 7-6(7-2)
 9 Thomas Johansson Marat Safin         3-6, 6-4, 6-4, 7-6(7-4)
```

- **Show the years for which prize data is available.**

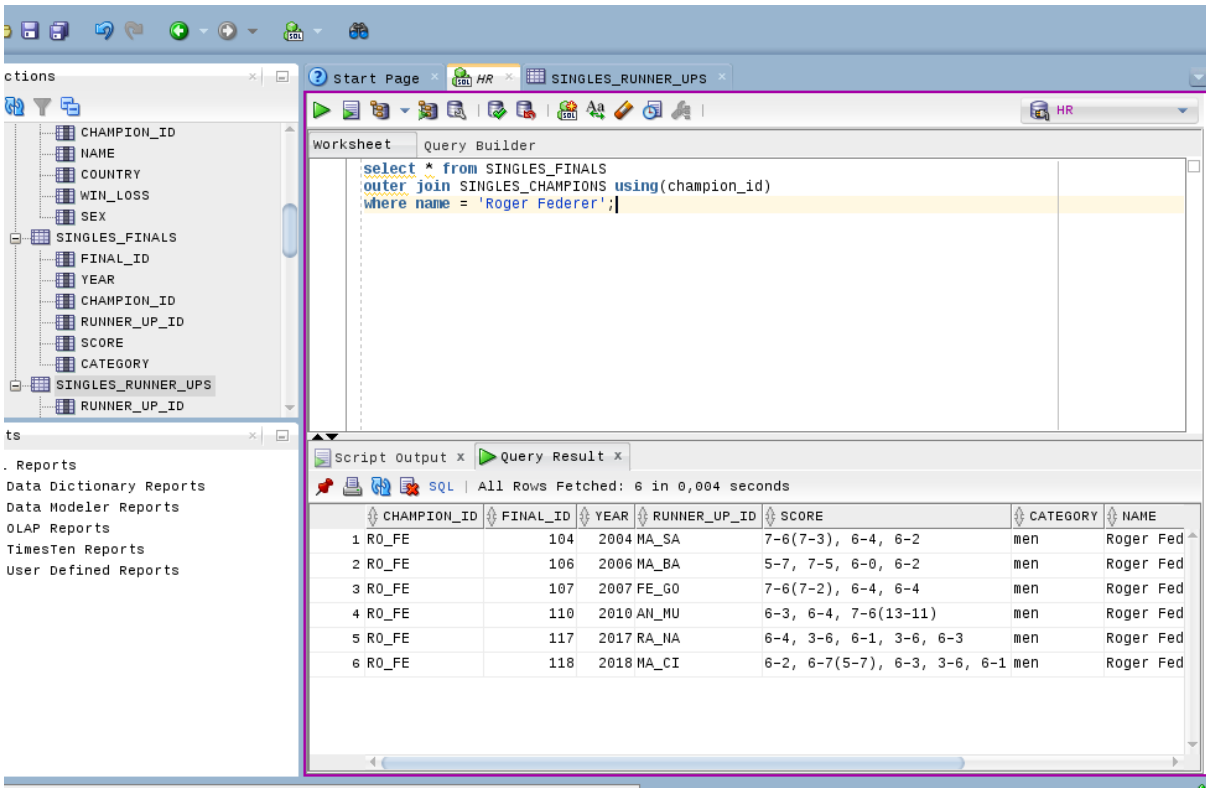


```
select year from SINGLES_FINALS
outer join prize using(final_id);
```

All Rows Fetched: 21 in 0,004 seconds

```
     YEAR
 1   2010
 2   2011
 3   2012
 4   2013
 5   2014
 6   2015
 7   2016
 8   2017
 9   2018
10   2019
```

- **Display only the finals won by Roger Federer.**

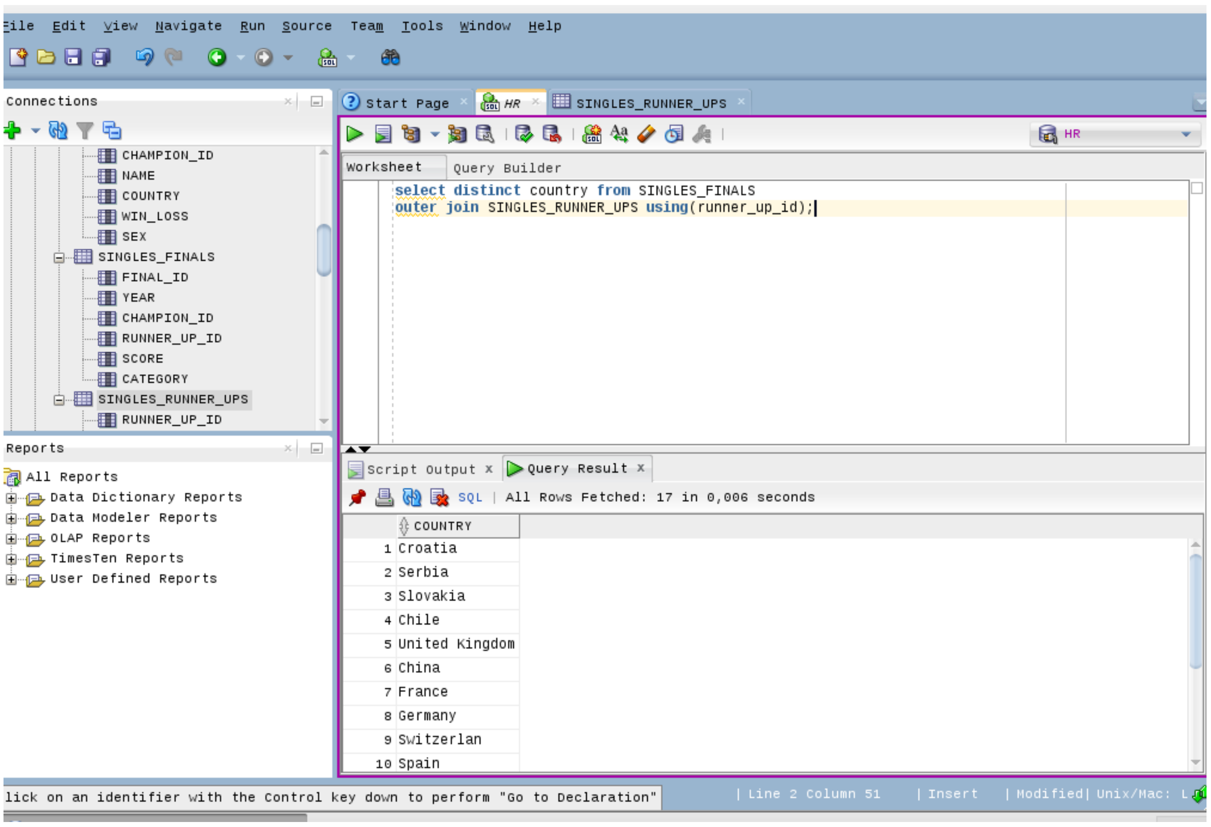


```
select * from SINGLES_FINALS
outer join SINGLES_CHAMPIONS using(champion_id)
where name = 'Roger Federer';
```

All Rows Fetched: 6 in 0,004 seconds

	CHAMPION_ID	FINAL_ID	YEAR	RUNNER_UP_ID	SCORE	CATEGORY	NAME
1	RO_FE	104	2004	MA_SA	7-6(7-3), 6-4, 6-2	men	Roger Fed
2	RO_FE	106	2006	MA_BA	5-7, 7-5, 6-0, 6-2	men	Roger Fed
3	RO_FE	107	2007	FE_GO	7-6(7-2), 6-4, 6-4	men	Roger Fed
4	RO_FE	110	2010	AN_MU	6-3, 6-4, 7-6(13-11)	men	Roger Fed
5	RO_FE	117	2017	RA_NA	6-4, 3-6, 6-1, 3-6, 6-3	men	Roger Fed
6	RO_FE	118	2018	MA_CI	6-2, 6-7(5-7), 6-3, 3-6, 6-1	men	Roger Fed

- **List the countries with runners-up.**



```
CHAMPION_ID
NAME
COUNTRY
WIN_LOSS
SEX
SINGLES_FINALS
FINAL_ID
YEAR
CHAMPION_ID
RUNNER_UP_ID
SCORE
CATEGORY
SINGLES_RUNNER_UPS
RUNNER_UP_ID
```

```
select distinct country from SINGLES_FINALS
outer join SINGLES_RUNNER_UPS using(runner_up_id);
```

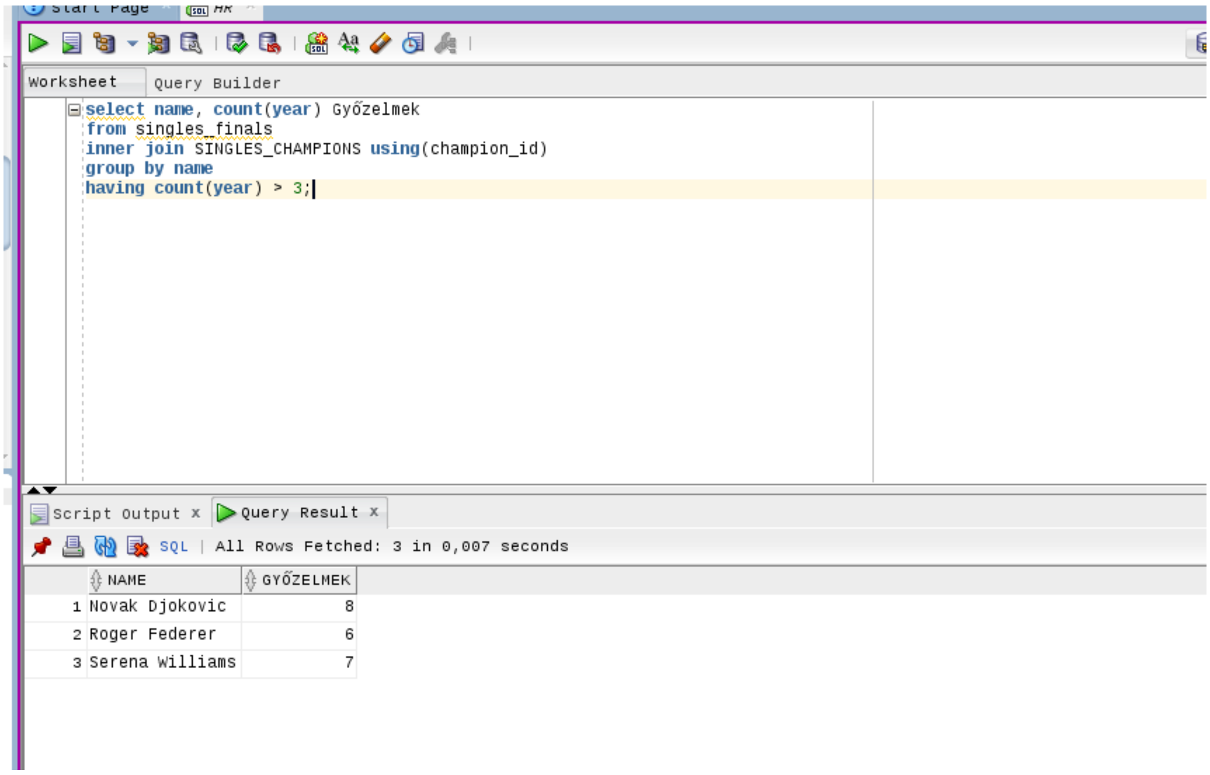
All Rows Fetched: 17 in 0,006 seconds

	COUNTRY
1	Croatia
2	Serbia
3	Slovakia
4	Chile
5	United Kingdom
6	China
7	France
8	Germany
9	Switzerlan
10	Spain

lick on an identifier with the Control key down to perform "Go to Declaration" | Line 2 Column 51 | Insert | Modified| Unix/Mac: L

7. Grouping Queries

- **List players who have won more than three times,** along with their win counts, sorted by the number of wins in descending order.

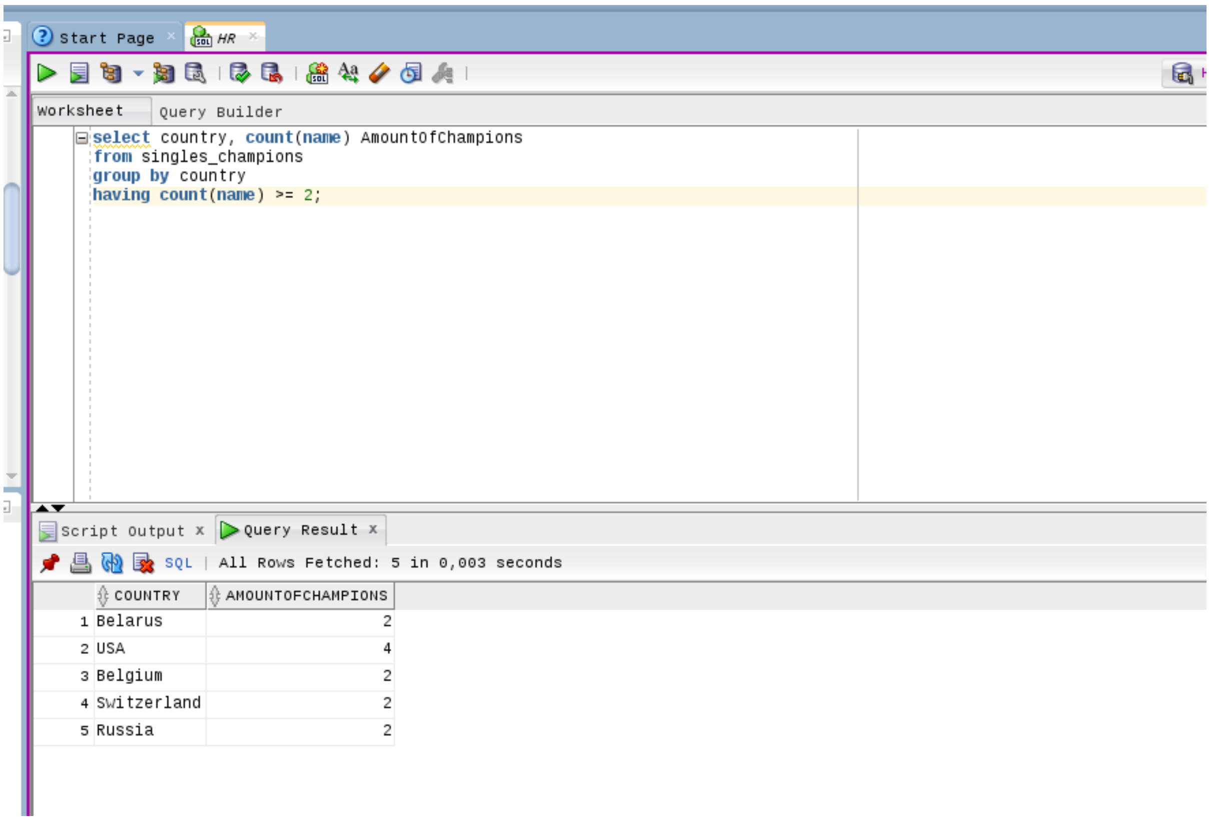


```
select name, count(year) Győzelmek
from singles_finals
inner join SINGLES_CHAMPIONS using(champion_id)
group by name
having count(year) > 3;
```

```
All Rows Fetched: 3 in 0,007 seconds
```

	NAME	GYŐZELMEK
1	Novak Djokovic	8
2	Roger Federer	6
3	Serena Williams	7

- **Show the countries with more than two champions,** along with the number of champions from each country.

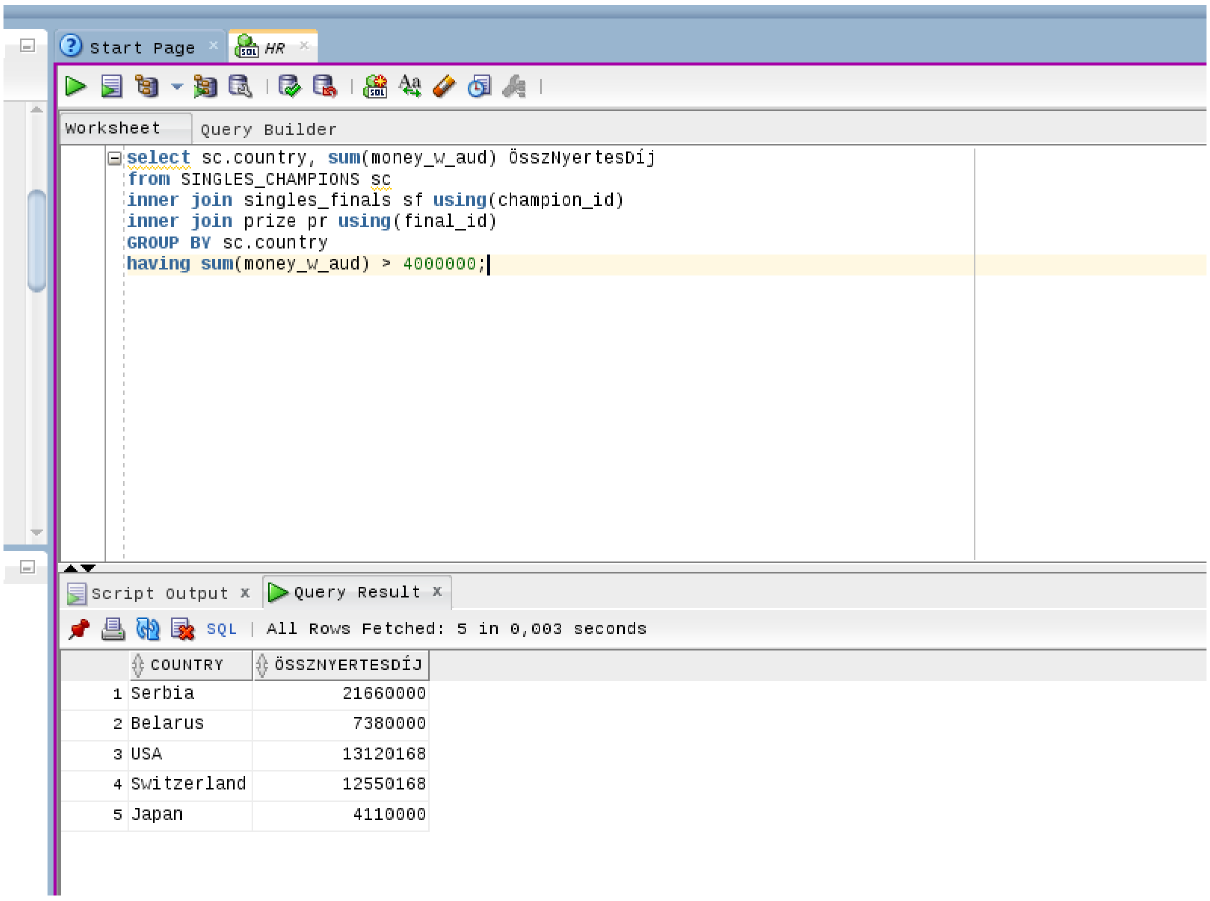


```
select country, count(name) AmountOfChampions
from singles_champions
group by country
having count(name) >= 2;
```

All Rows Fetched: 5 in 0,003 seconds

	COUNTRY	AMOUNTOFCHAMPIONS
1	Belarus	2
2	USA	4
3	Belgium	2
4	Switzerland	2
5	Russia	2

- **Display the total prize money earned by champions for each country** where the amount exceeds AUD 4,000,000.



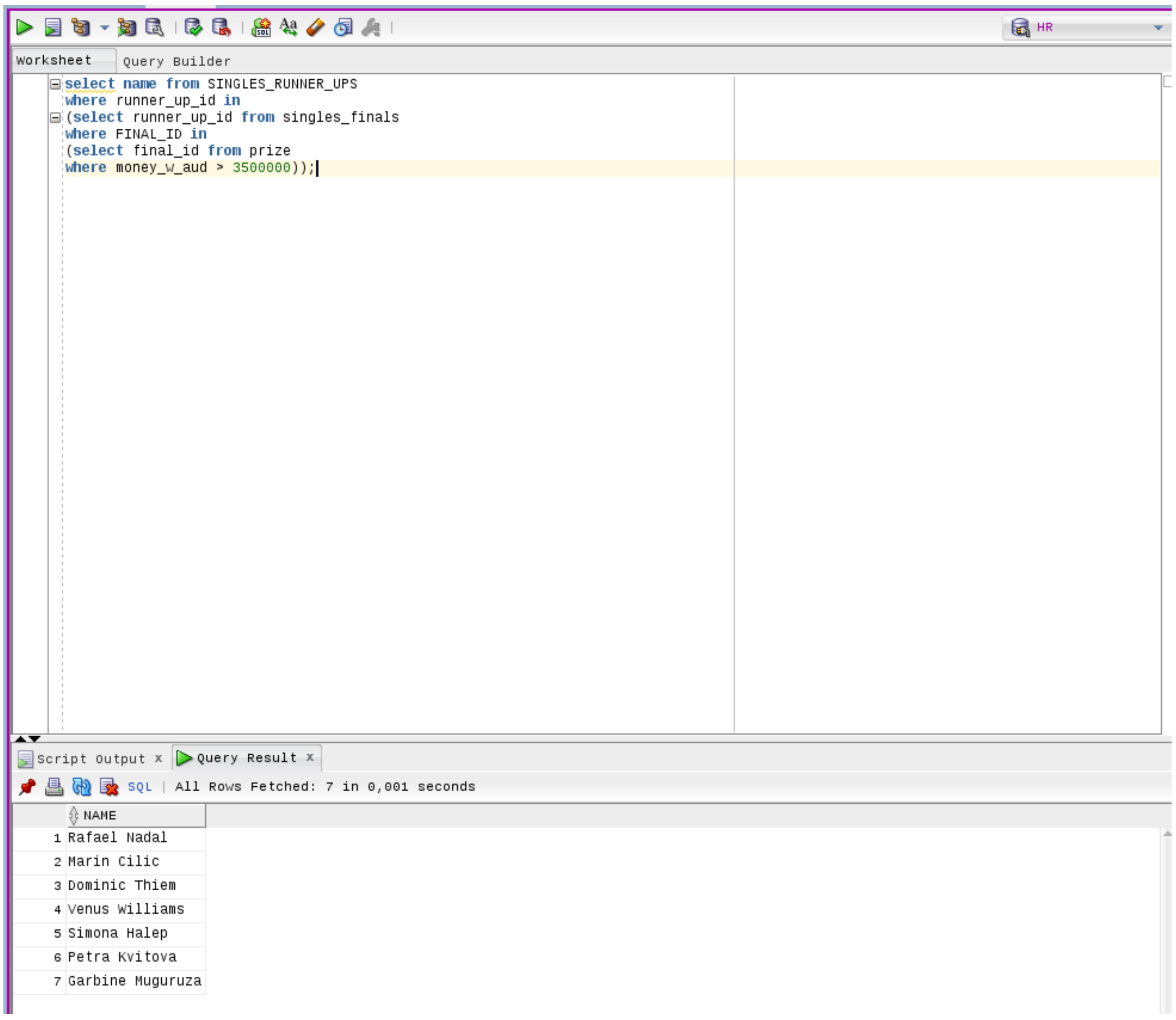
```
select sc.country, sum(money_w_aud) összNyertesDíj
from SINGLES_CHAMPIONS sc
inner join singles_finals sf using(champion_id)
inner join prize pr using(final_id)
GROUP BY sc.country
having sum(money_w_aud) > 4000000;
```

All Rows Fetched: 5 in 0,003 seconds

	COUNTRY	ÖSSZNYERTESDÍJ
1	Serbia	21660000
2	Belarus	7380000
3	USA	13120168
4	Switzerland	12550168
5	Japan	4110000

8. Subqueries

- **List players who finished as runners-up** in tournaments with a first-place prize exceeding AUD 3,500,000.



Worksheet | Query Builder

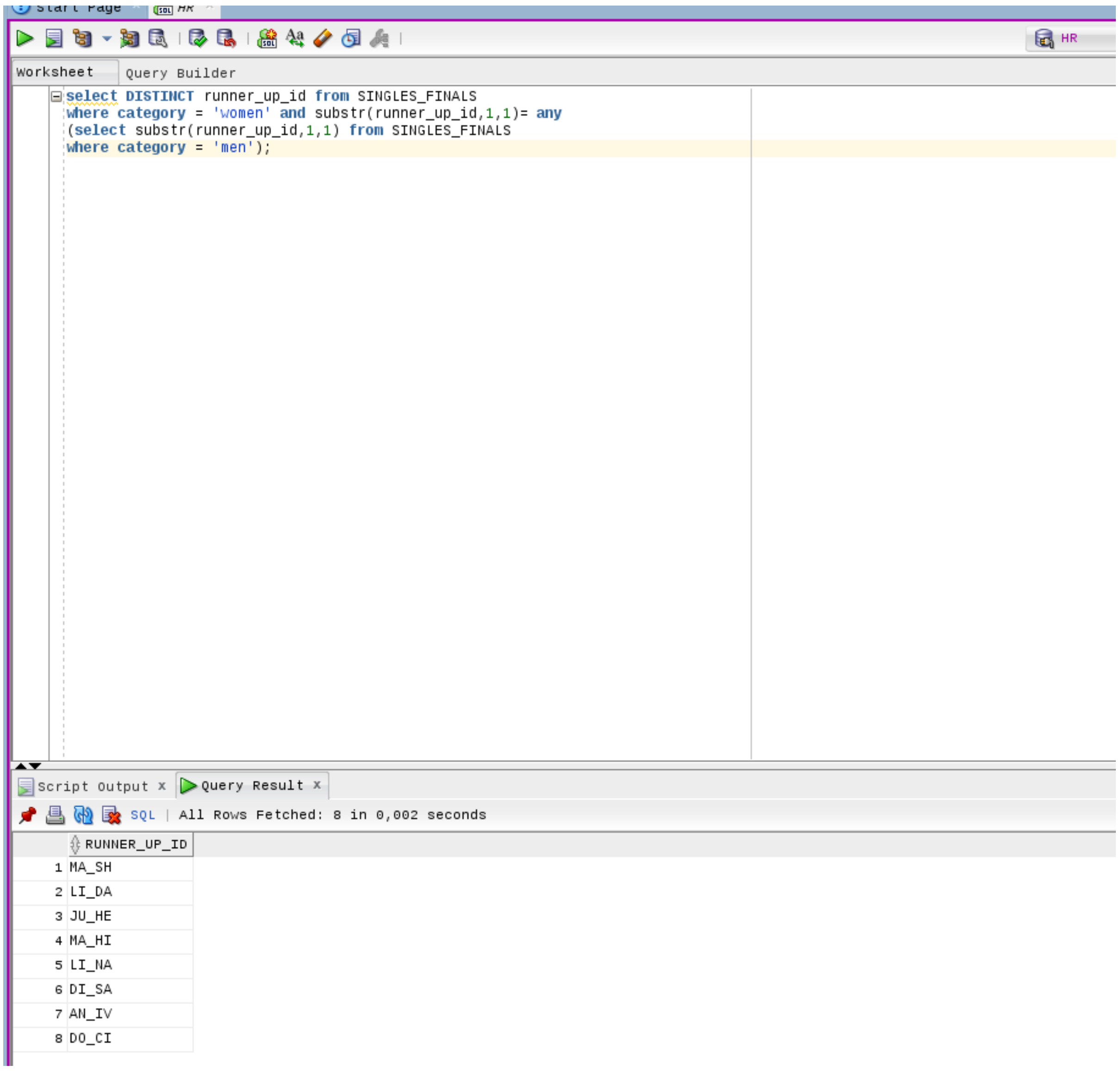
```sql
select name from SINGLES_RUNNER_UPS
where runner_up_id in
(select runner_up_id from singles_finals
where FINAL_ID in
(select final_id from prize
where money_w_aud > 3500000));
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 7 in 0,001 seconds

	NAME
1	Rafael Nadal
2	Marin Cilic
3	Dominic Thiem
4	Venus Williams
5	Simona Halep
6	Petra Kvitova
7	Garbine Muguruza

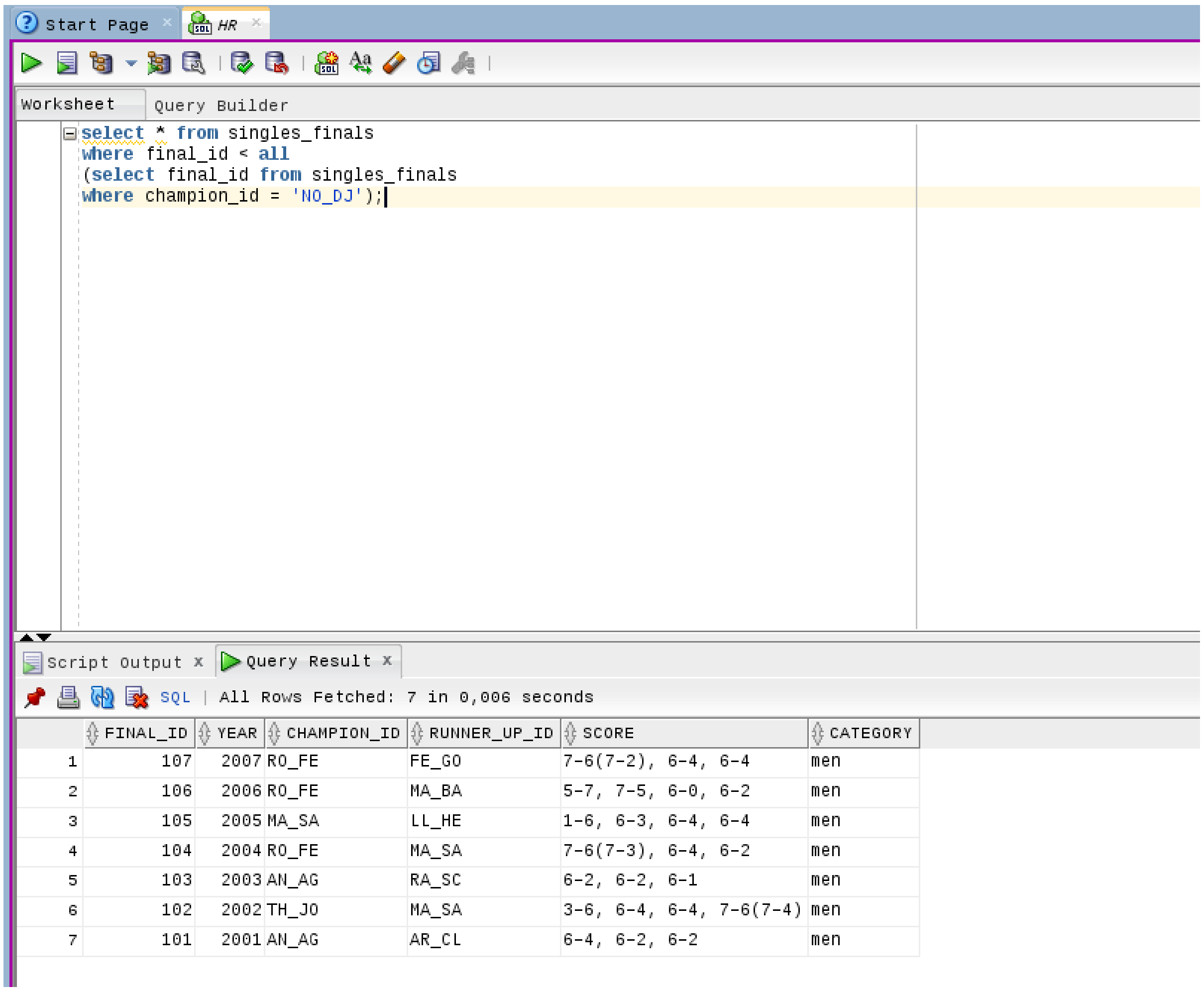
- **List female runners-up** whose first name initial matches the initial of any male runner-up's first name.



```
select DISTINCT runner_up_id from SINGLES_FINALS
where category = 'women' and substr(runner_up_id,1,1)= any
(select substr(runner_up_id,1,1) from SINGLES_FINALS
where category = 'men');
```

```
All Rows Fetched: 8 in 0,002 seconds

  RUNNER_UP_ID
1 MA_SH
2 LI_DA
3 JU_HE
4 MA_HI
5 LI_NA
6 DI_SA
7 AN_IV
8 DO_CI
```

- **Show all finals that occurred before Novak Djokovic's first tournament victory.**

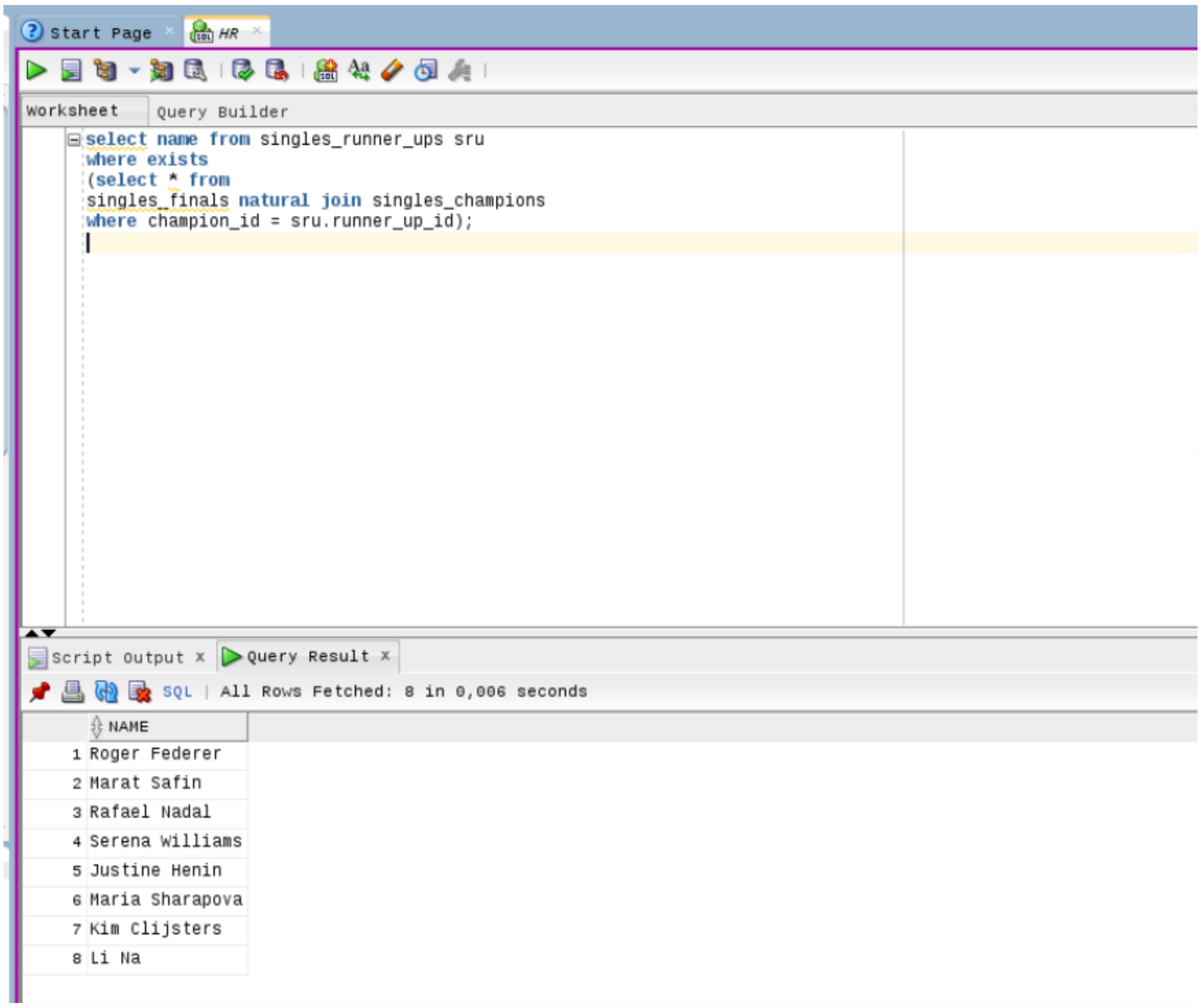


```
select * from singles_finals
where final_id < all
(select final_id from singles_finals
where champion_id = 'NO_DJ');
```

All Rows Fetched: 7 in 0,006 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	SCORE	CATEGORY
1	107	2007	RO_FE	FE_GO	7-6(7-2), 6-4, 6-4	men
2	106	2006	RO_FE	MA_BA	5-7, 7-5, 6-0, 6-2	men
3	105	2005	MA_SA	LL_HE	1-6, 6-3, 6-4, 6-4	men
4	104	2004	RO_FE	MA_SA	7-6(7-3), 6-4, 6-2	men
5	103	2003	AN_AG	RA_SC	6-2, 6-2, 6-1	men
6	102	2002	TH_JO	MA_SA	3-6, 6-4, 6-4, 7-6(7-4)	men
7	101	2001	AN_AG	AR_CL	6-4, 6-2, 6-2	men

- **List players who have achieved both first and second places** in the tournament.

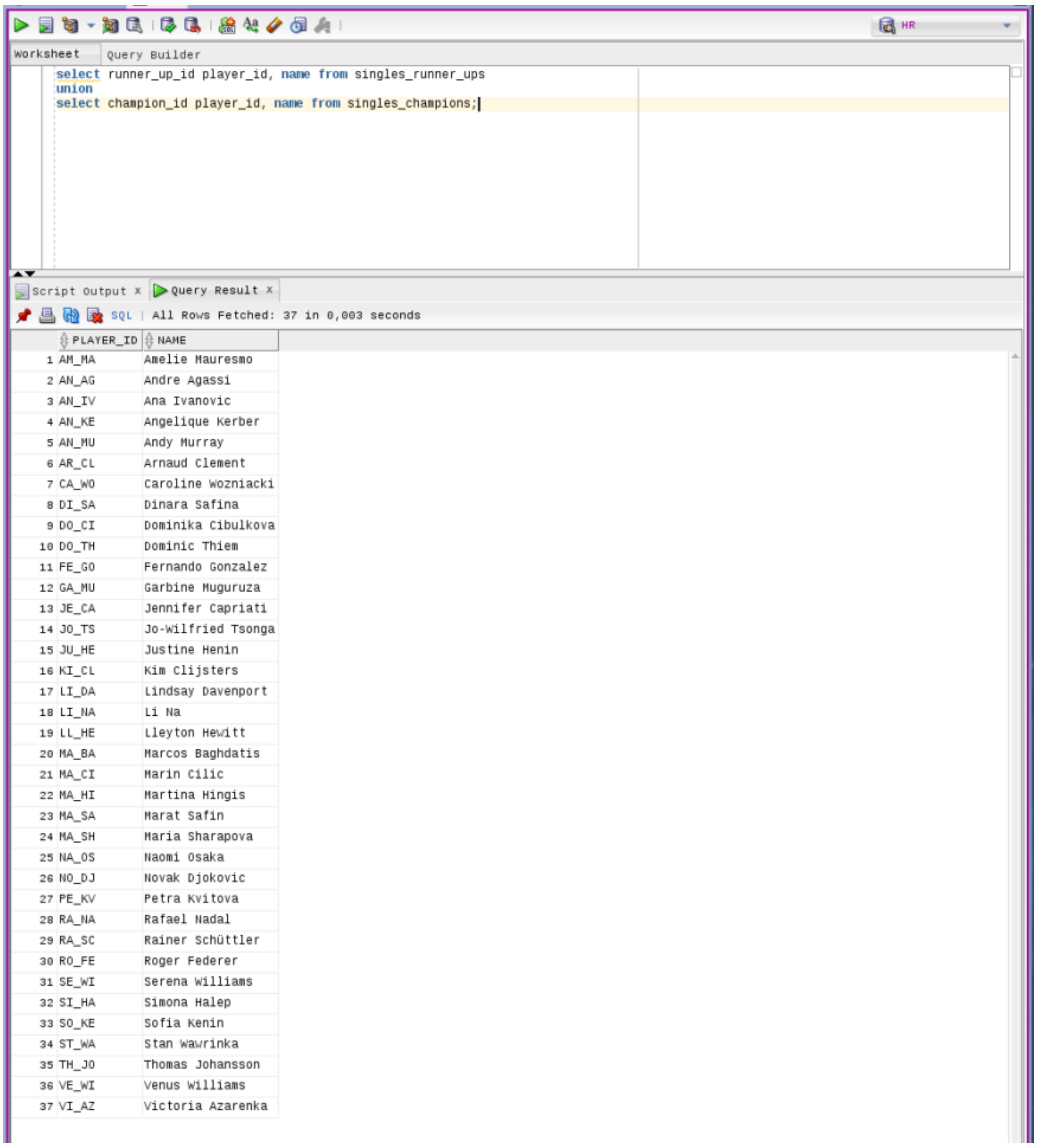


```
select name from singles_runner_ups sru
where exists
(select * from
singles_finals natural join singles_champions
where champion_id = sru.runner_up_id);
```

```
All Rows Fetched: 8 in 0,006 seconds

    NAME
  1 Roger Federer
  2 Marat Safin
  3 Rafael Nadal
  4 Serena Williams
  5 Justine Henin
  6 Maria Sharapova
  7 Kim Clijsters
  8 Li Na
```

9. Set Operations

- **List all finalists' IDs and names.**

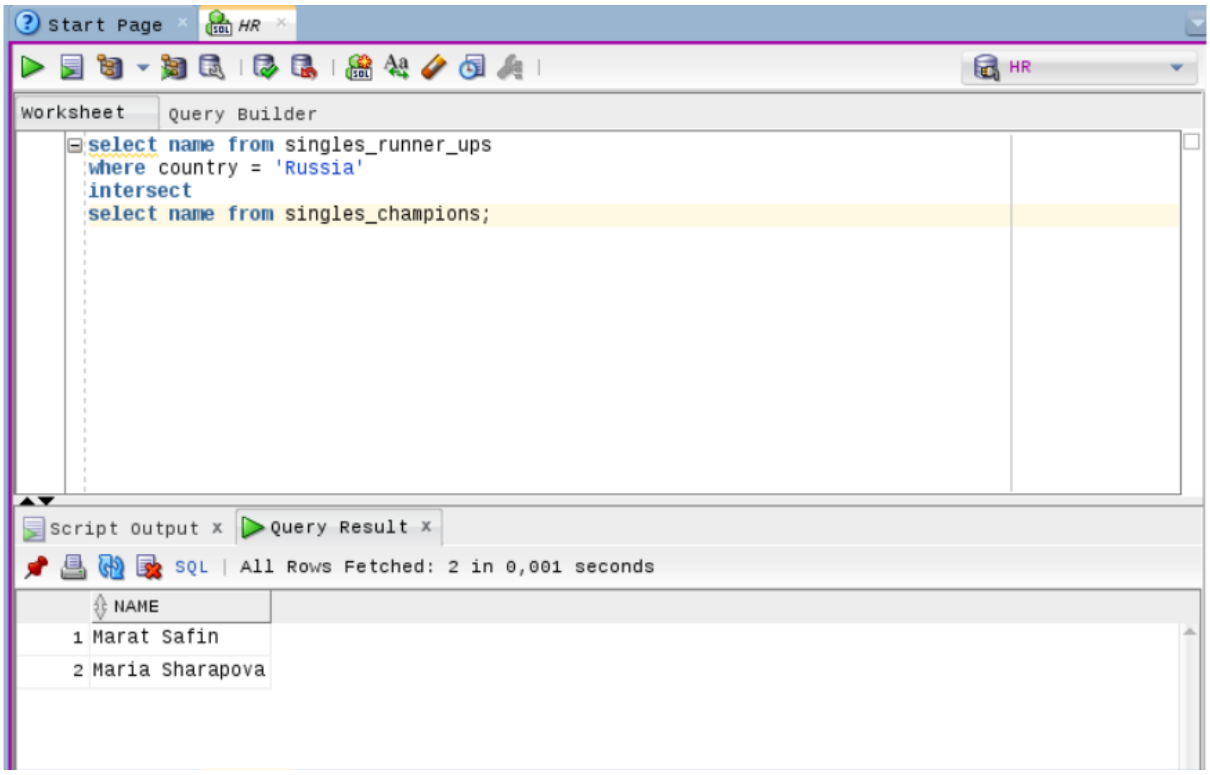


```
select runner_up_id player_id, name from singles_runner_ups
union
select champion_id player_id, name from singles_champions;
```

```
All Rows Fetched: 37 in 0,003 seconds
```

	PLAYER_ID	NAME
1	AM_MA	Amelie Mauresmo
2	AN_AG	Andre Agassi
3	AN_IV	Ana Ivanovic
4	AN_KE	Angelique Kerber
5	AN_MU	Andy Murray
6	AR_CL	Arnaud Clement
7	CA_WO	Caroline Wozniacki
8	DI_SA	Dinara Safina
9	DO_CI	Dominika Cibulkova
10	DO_TH	Dominic Thiem
11	FE_GO	Fernando Gonzalez
12	GA_MU	Garbine Muguruza
13	JE_CA	Jennifer Capriati
14	JO_TS	Jo-Wilfried Tsonga
15	JU_HE	Justine Henin
16	KI_CL	Kim Clijsters
17	LI_DA	Lindsay Davenport
18	LI_NA	Li Na
19	LL_HE	Lleyton Hewitt
20	MA_BA	Marcos Baghdatis
21	MA_CI	Marin Cilic
22	MA_HI	Martina Hingis
23	MA_SA	Marat Safin
24	MA_SH	Maria Sharapova
25	NA_OS	Naomi Osaka
26	NO_DJ	Novak Djokovic
27	PE_KV	Petra Kvitova
28	RA_NA	Rafael Nadal
29	RA_SC	Rainer Schüttler
30	RO_FE	Roger Federer
31	SE_WI	Serena Williams
32	SI_HA	Simona Halep
33	SO_KE	Sofia Kenin
34	ST_WA	Stan Wawrinka
35	TH_JO	Thomas Johansson
36	VE_WI	Venus Williams
37	VI_AZ	Victoria Azarenka

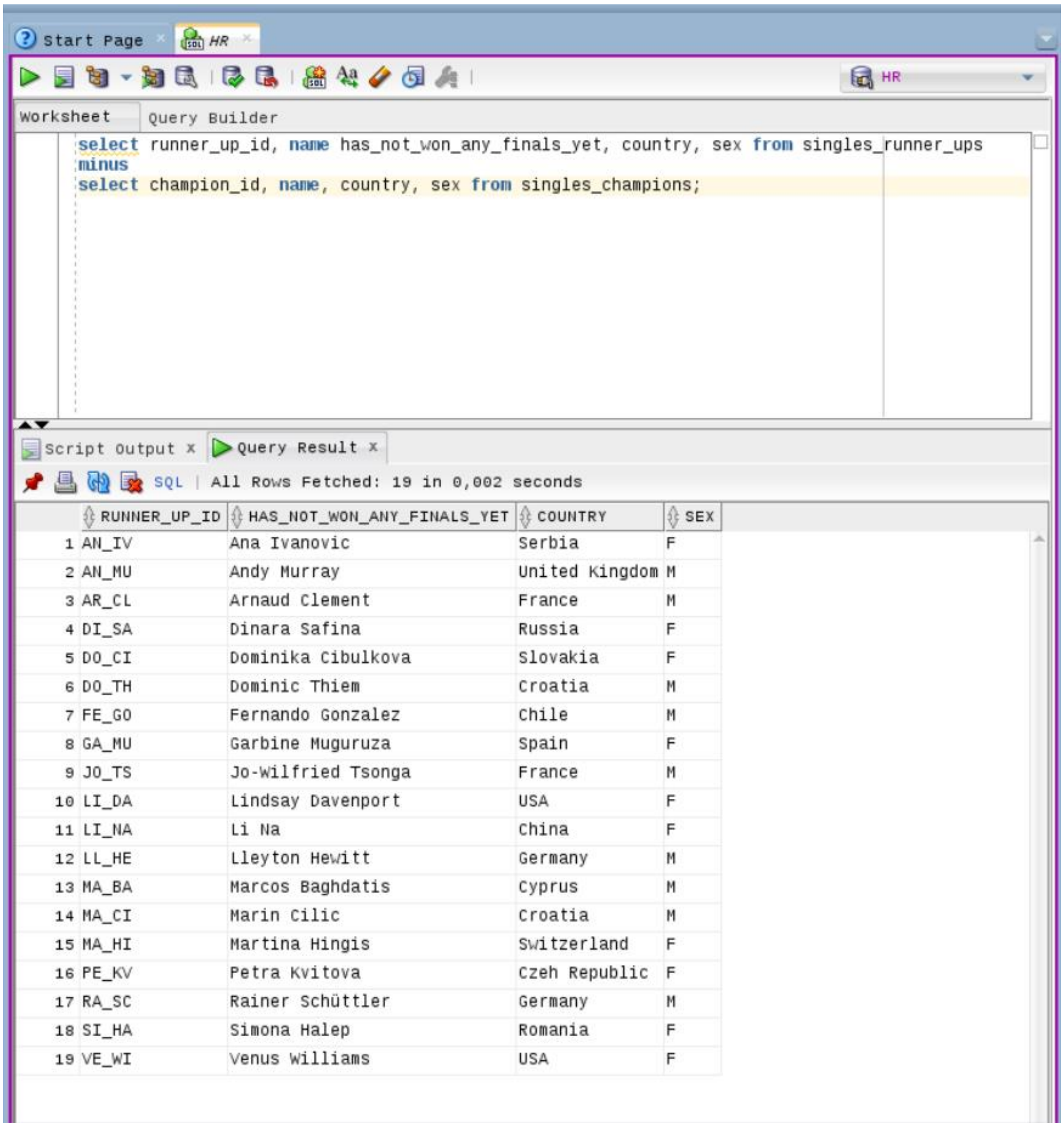
- **Select players who are Russian and have achieved both first and second places.**



```
select name from singles_runner_ups
where country = 'Russia'
intersect
select name from singles_champions;
```

```
SQL | All Rows Fetched: 2 in 0,001 seconds

  NAME
1 Marat Safin
2 Maria Sharapova
```

- **Identify players who have never won a final.**



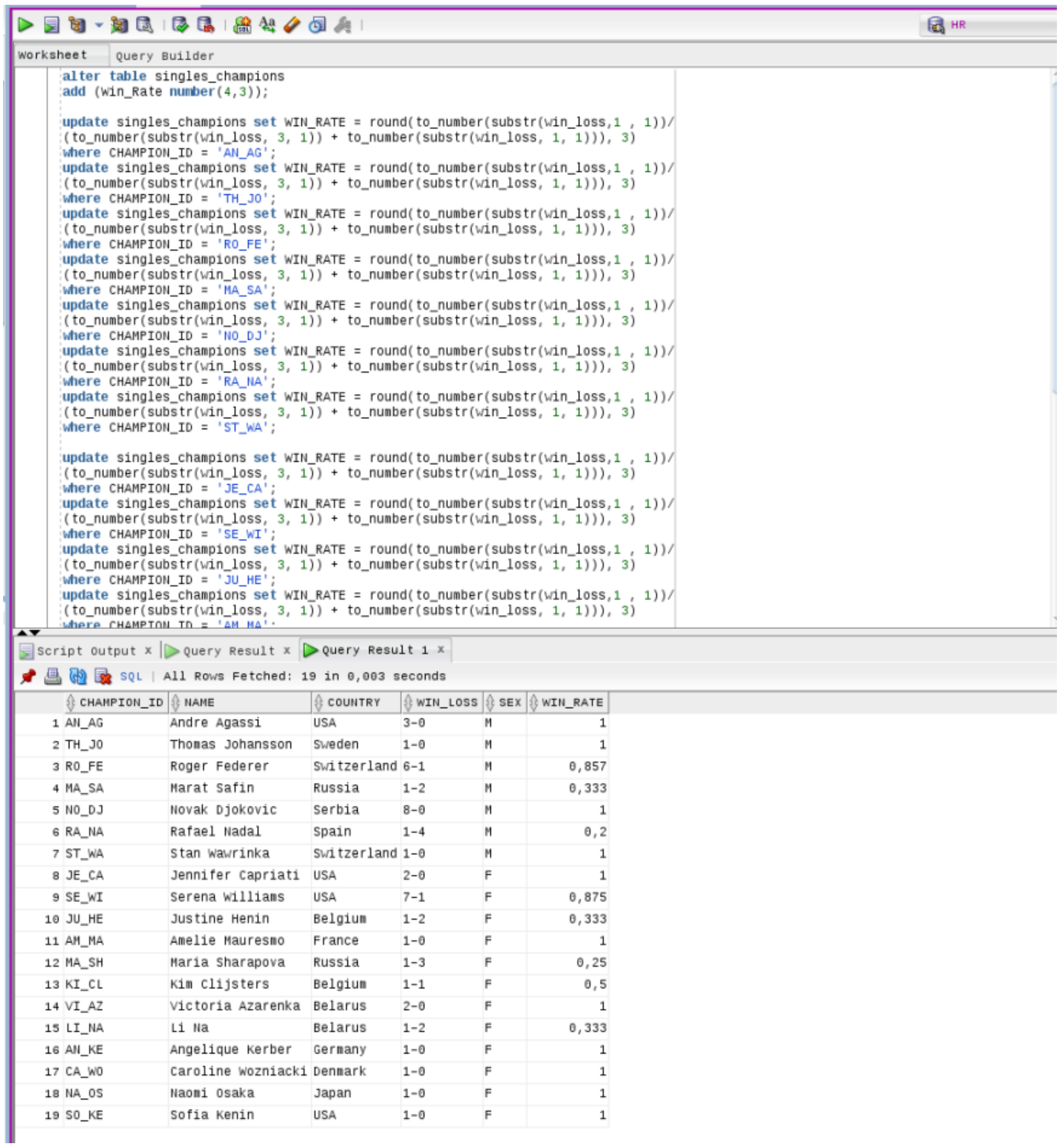
```
select runner_up_id, name has_not_won_any_finals_yet, country, sex from singles_runner_ups
minus
select champion_id, name, country, sex from singles_champions;
```

All Rows Fetched: 19 in 0,002 seconds

	RUNNER_UP_ID	HAS_NOT_WON_ANY_FINALS_YET	COUNTRY	SEX
1	AN_IV	Ana Ivanovic	Serbia	F
2	AN_MU	Andy Murray	United Kingdom	M
3	AR_CL	Arnaud Clement	France	M
4	DI_SA	Dinara Safina	Russia	F
5	DO_CI	Dominika Cibulkova	Slovakia	F
6	DO_TH	Dominic Thiem	Croatia	M
7	FE_GO	Fernando Gonzalez	Chile	M
8	GA_MU	Garbine Muguruza	Spain	F
9	JO_TS	Jo-Wilfried Tsonga	France	M
10	LI_DA	Lindsay Davenport	USA	F
11	LI_NA	Li Na	China	F
12	LL_HE	Lleyton Hewitt	Germany	M
13	MA_BA	Marcos Baghdatis	Cyprus	M
14	MA_CI	Marin Cilic	Croatia	M
15	MA_HI	Martina Hingis	Switzerland	F
16	PE_KV	Petra Kvitova	Czeh Republic	F
17	RA_SC	Rainer Schüttler	Germany	M
18	SI_HA	Simona Halep	Romania	F
19	VE_WI	Venus Williams	USA	F

10. Views, DML, and DDL Operations

- **Add a new "Win Rate" column** to the Singles Champions table and populate it with relevant data.



Worksheet | Query Builder

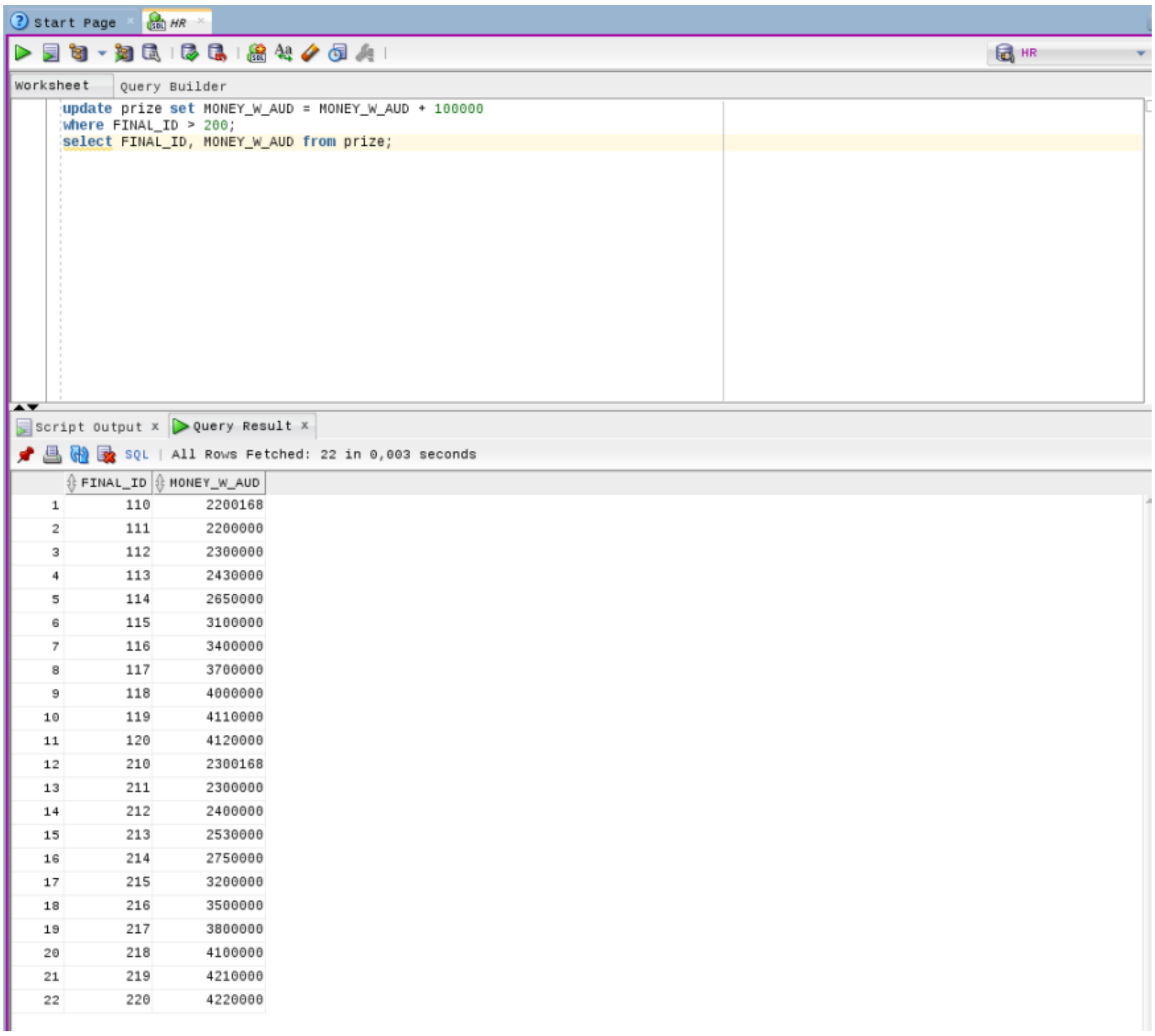
```
alter table singles_champions
add (win_Rate number(4,3));

update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'AN_AG';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'TH_JO';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'RO_FE';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'MA_SA';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'NO_DJ';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'RA_NA';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'ST_WA';

update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'JE_CA';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'SE_WI';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'JU_HE';
update singles_champions set WIN_RATE = round(to_number(substr(win_loss,1 , 1))/
(to_number(substr(win_loss, 3, 1)) + to_number(substr(win_loss, 1, 1))), 3)
where CHAMPION_ID = 'AM_MA';
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 19 in 0,003 seconds

	CHAMPION_ID	NAME	COUNTRY	WIN_LOSS	SEX	WIN_RATE
1	AN_AG	Andre Agassi	USA	3-0	M	1
2	TH_JO	Thomas Johansson	Sweden	1-0	M	1
3	RO_FE	Roger Federer	Switzerland	6-1	M	0,857
4	MA_SA	Marat Safin	Russia	1-2	M	0,333
5	NO_DJ	Novak Djokovic	Serbia	8-0	M	1
6	RA_NA	Rafael Nadal	Spain	1-4	M	0,2
7	ST_WA	Stan Wawrinka	Switzerland	1-0	M	1
8	JE_CA	Jennifer Capriati	USA	2-0	F	1
9	SE_WI	Serena Williams	USA	7-1	F	0,875
10	JU_HE	Justine Henin	Belgium	1-2	F	0,333
11	AM_MA	Amelie Mauresmo	France	1-0	F	1
12	MA_SH	Maria Sharapova	Russia	1-3	F	0,25
13	KI_CL	Kim Clijsters	Belgium	1-1	F	0,5
14	VI_AZ	Victoria Azarenka	Belarus	2-0	F	1
15	LI_NA	Li Na	Belarus	1-2	F	0,333
16	AN_KE	Angelique Kerber	Germany	1-0	F	1
17	CA_WO	Caroline Wozniacki	Denmark	1-0	F	1
18	NA_OS	Naomi Osaka	Japan	1-0	F	1
19	SO_KE	Sofia Kenin	USA	1-0	F	1

- **Increase the winners' prize** by AUD 100,000 for women's finals (where Final ID > 200) every year.

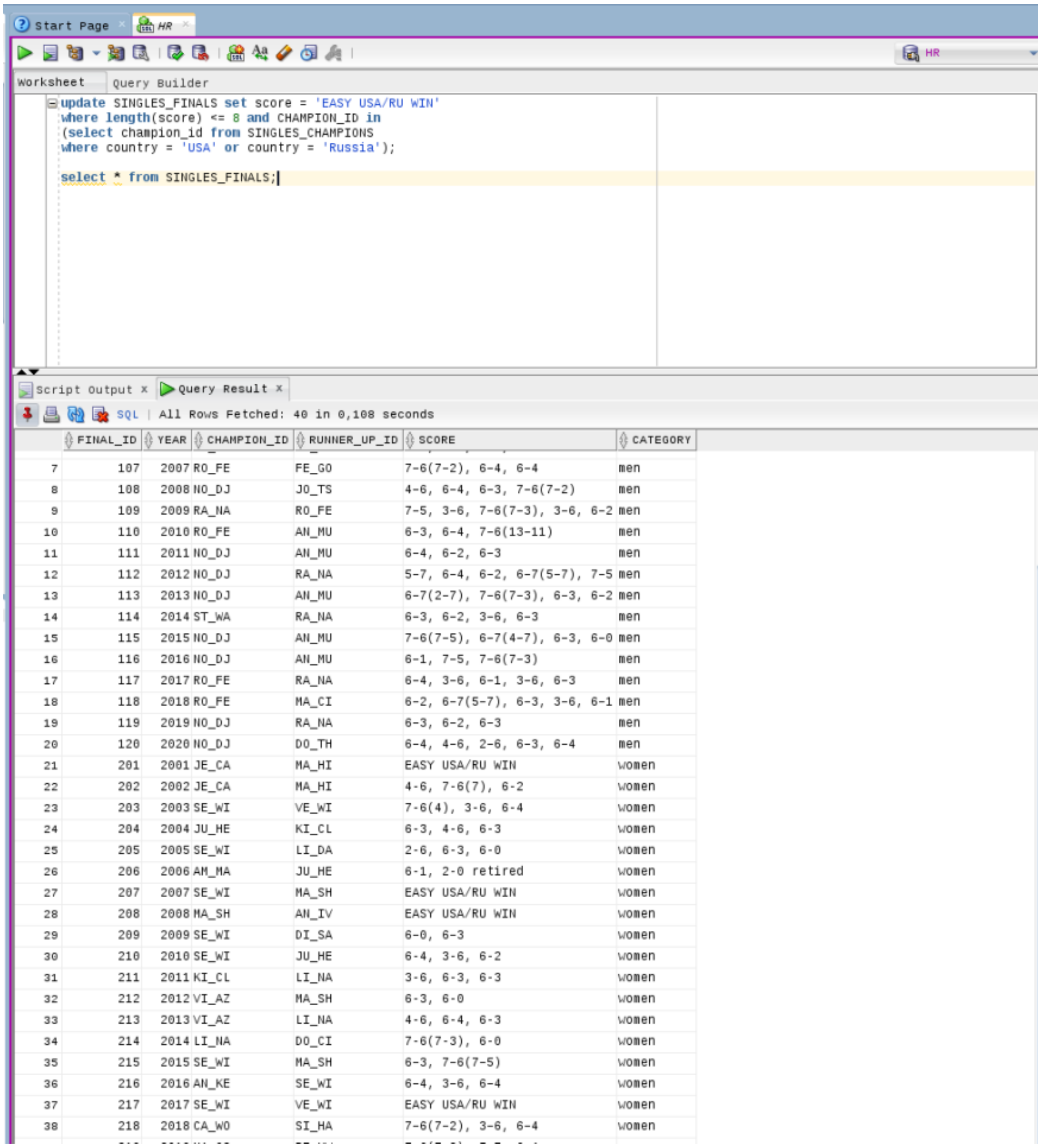


```sql
update prize set MONEY_W_AUD = MONEY_W_AUD + 100000
where FINAL_ID > 200;
select FINAL_ID, MONEY_W_AUD from prize;
```

All Rows Fetched: 22 in 0,003 seconds

	FINAL_ID	MONEY_W_AUD
1	110	2200168
2	111	2200000
3	112	2300000
4	113	2430000
5	114	2650000
6	115	3100000
7	116	3400000
8	117	3700000
9	118	4000000
10	119	4110000
11	120	4120000
12	210	2300168
13	211	2300000
14	212	2400000
15	213	2530000
16	214	2750000
17	215	3200000
18	216	3500000
19	217	3800000
20	218	4100000
21	219	4210000
22	220	4220000

- **Replace the score result with "EASY USA/RU WIN"** for matches won in two sets without a Tie-Break, where the winner was American or Russian.

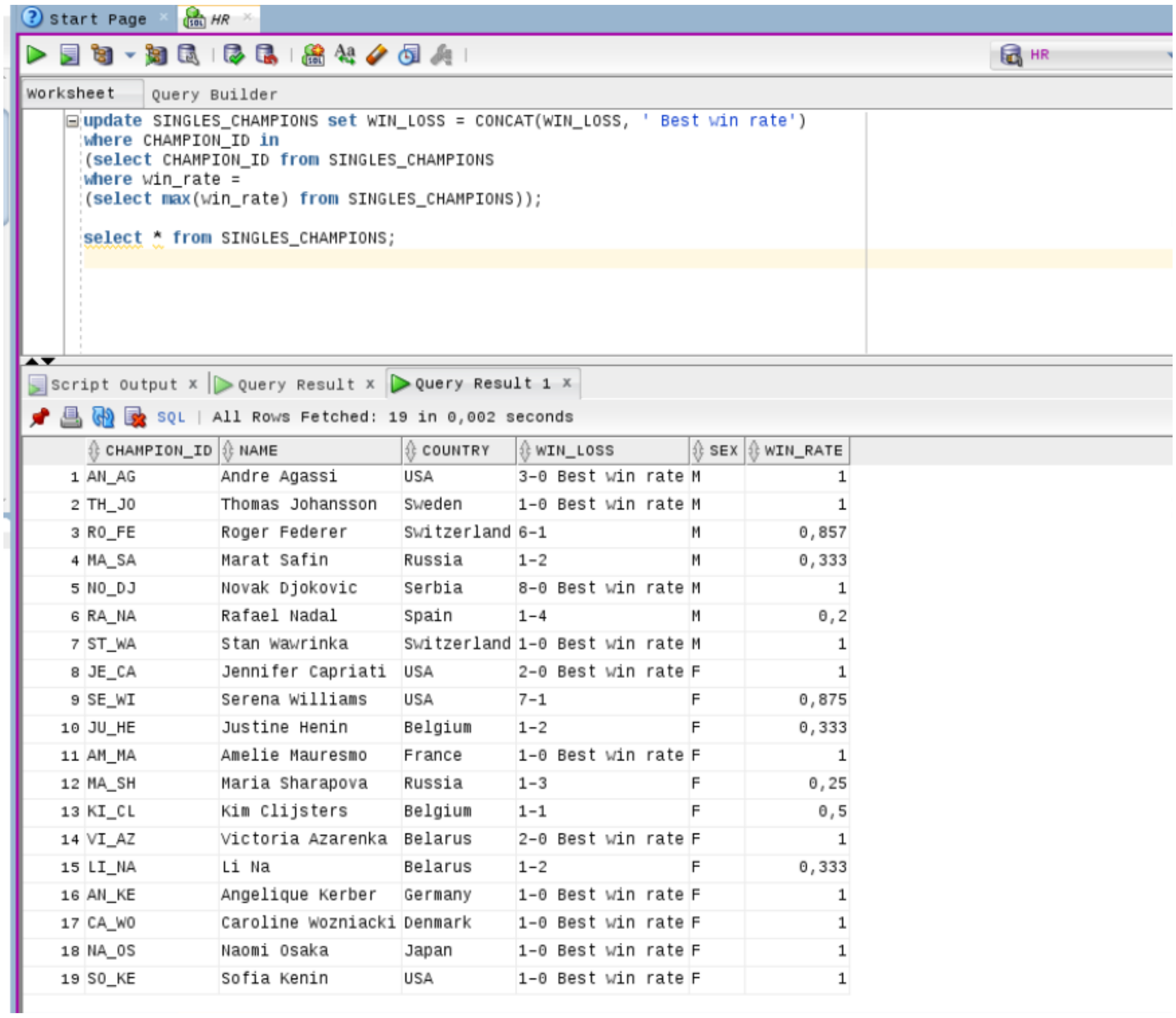


```sql
update SINGLES_FINALS set score = 'EASY USA/RU WIN'
where length(score) <= 8 and CHAMPION_ID in
(select champion_id from SINGLES_CHAMPIONS
where country = 'USA' or country = 'Russia');

select * from SINGLES_FINALS;
```

All Rows Fetched: 40 in 0,108 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	SCORE	CATEGORY
7	107	2007	RO_FE	FE_GO	7-6(7-2), 6-4, 6-4	men
8	108	2008	NO_DJ	JO_TS	4-6, 6-4, 6-3, 7-6(7-2)	men
9	109	2009	RA_NA	RO_FE	7-5, 3-6, 7-6(7-3), 3-6, 6-2	men
10	110	2010	RO_FE	AN_MU	6-3, 6-4, 7-6(13-11)	men
11	111	2011	NO_DJ	AN_MU	6-4, 6-2, 6-3	men
12	112	2012	NO_DJ	RA_NA	5-7, 6-4, 6-2, 6-7(5-7), 7-5	men
13	113	2013	NO_DJ	AN_MU	6-7(2-7), 7-6(7-3), 6-3, 6-2	men
14	114	2014	ST_WA	RA_NA	6-3, 6-2, 3-6, 6-3	men
15	115	2015	NO_DJ	AN_MU	7-6(7-5), 6-7(4-7), 6-3, 6-0	men
16	116	2016	NO_DJ	AN_MU	6-1, 7-5, 7-6(7-3)	men
17	117	2017	RO_FE	RA_NA	6-4, 3-6, 6-1, 3-6, 6-3	men
18	118	2018	RO_FE	MA_CI	6-2, 6-7(5-7), 6-3, 3-6, 6-1	men
19	119	2019	NO_DJ	RA_NA	6-3, 6-2, 6-3	men
20	120	2020	NO_DJ	DO_TH	6-4, 4-6, 2-6, 6-3, 6-4	men
21	201	2001	JE_CA	MA_HI	EASY USA/RU WIN	women
22	202	2002	JE_CA	MA_HI	4-6, 7-6(7), 6-2	women
23	203	2003	SE_WI	VE_WI	7-6(4), 3-6, 6-4	women
24	204	2004	JU_HE	KI_CL	6-3, 4-6, 6-3	women
25	205	2005	SE_WI	LI_DA	2-6, 6-3, 6-0	women
26	206	2006	AM_MA	JU_HE	6-1, 2-0 retired	women
27	207	2007	SE_WI	MA_SH	EASY USA/RU WIN	women
28	208	2008	MA_SH	AN_IV	EASY USA/RU WIN	women
29	209	2009	SE_WI	DI_SA	6-0, 6-3	women
30	210	2010	SE_WI	JU_HE	6-4, 3-6, 6-2	women
31	211	2011	KI_CL	LI_NA	3-6, 6-3, 6-3	women
32	212	2012	VI_AZ	MA_SH	6-3, 6-0	women
33	213	2013	VI_AZ	LI_NA	4-6, 6-4, 6-3	women
34	214	2014	LI_NA	DO_CI	7-6(7-3), 6-0	women
35	215	2015	SE_WI	MA_SH	6-3, 7-6(7-5)	women
36	216	2016	AN_KE	SE_WI	6-4, 3-6, 6-4	women
37	217	2017	SE_WI	VE_WI	EASY USA/RU WIN	women
38	218	2018	CA_WO	SI_HA	7-6(7-2), 3-6, 6-4	women

- **Mark players with the best win ratio** by appending "Best win rate" to the Win-Loss column.

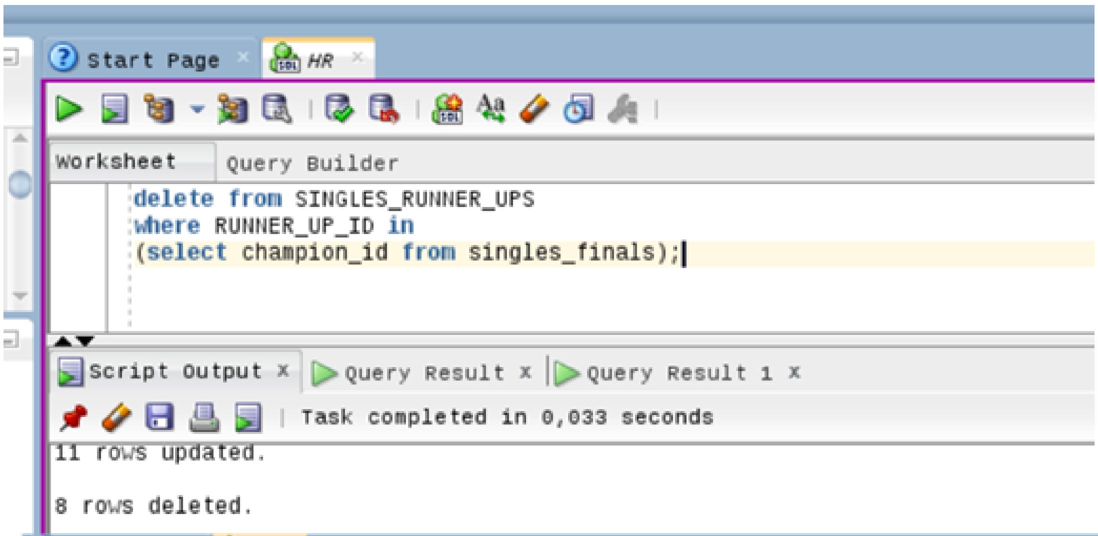


```
update SINGLES_CHAMPIONS set WIN_LOSS = CONCAT(WIN_LOSS, ' Best win rate')
where CHAMPION_ID in
(select CHAMPION_ID from SINGLES_CHAMPIONS
where win_rate =
(select max(win_rate) from SINGLES_CHAMPIONS));

select * from SINGLES_CHAMPIONS;
```

All Rows Fetched: 19 in 0,002 seconds

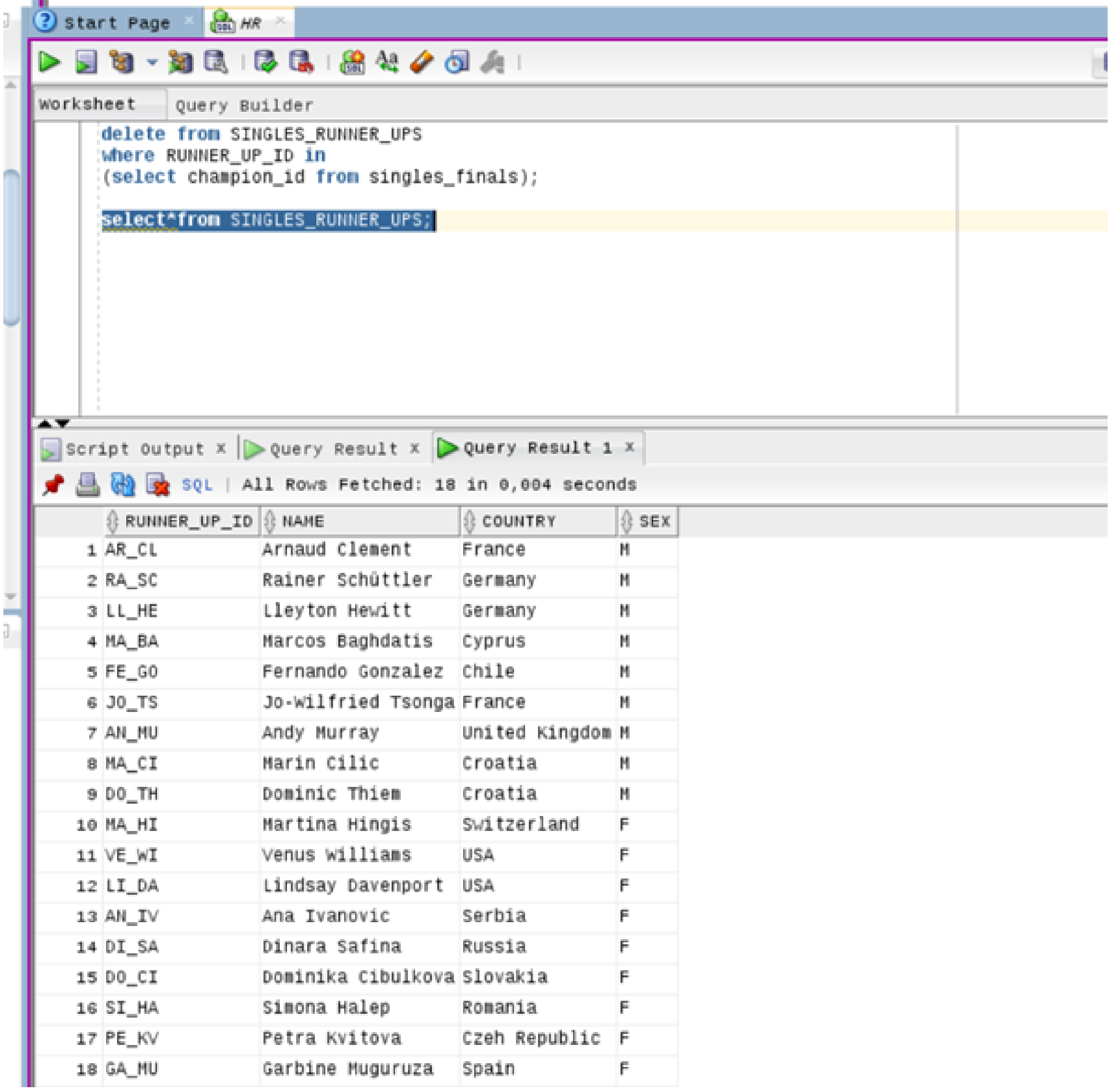
	CHAMPION_ID	NAME	COUNTRY	WIN_LOSS	SEX	WIN_RATE
1	AN_AG	Andre Agassi	USA	3-0 Best win rate	M	1
2	TH_JO	Thomas Johansson	Sweden	1-0 Best win rate	M	1
3	RO_FE	Roger Federer	Switzerland	6-1	M	0,857
4	MA_SA	Marat Safin	Russia	1-2	M	0,333
5	NO_DJ	Novak Djokovic	Serbia	8-0 Best win rate	M	1
6	RA_NA	Rafael Nadal	Spain	1-4	M	0,2
7	ST_WA	Stan Wawrinka	Switzerland	1-0 Best win rate	M	1
8	JE_CA	Jennifer Capriati	USA	2-0 Best win rate	F	1
9	SE_WI	Serena Williams	USA	7-1	F	0,875
10	JU_HE	Justine Henin	Belgium	1-2	F	0,333
11	AM_MA	Amelie Mauresmo	France	1-0 Best win rate	F	1
12	MA_SH	Maria Sharapova	Russia	1-3	F	0,25
13	KI_CL	Kim Clijsters	Belgium	1-1	F	0,5
14	VI_AZ	Victoria Azarenka	Belarus	2-0 Best win rate	F	1
15	LI_NA	Li Na	Belarus	1-2	F	0,333
16	AN_KE	Angelique Kerber	Germany	1-0 Best win rate	F	1
17	CA_WO	Caroline Wozniacki	Denmark	1-0 Best win rate	F	1
18	NA_OS	Naomi Osaka	Japan	1-0 Best win rate	F	1
19	SO_KE	Sofia Kenin	USA	1-0 Best win rate	F	1

- **Delete runners-up** from the Singles Runner-ups table who have ever won the tournament.



```
delete from SINGLES_RUNNER_UPS
where RUNNER_UP_ID in
(select champion_id from singles_finals);
```

```
Task completed in 0,033 seconds
11 rows updated.

8 rows deleted.
```

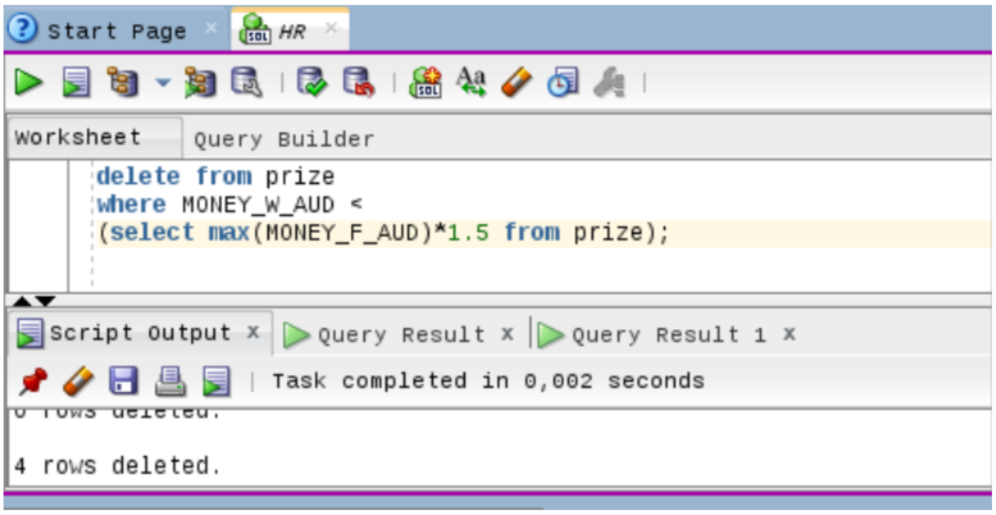


```
delete from SINGLES_RUNNER_UPS
where RUNNER_UP_ID in
(select champion_id from singles_finals);

select*from SINGLES_RUNNER_UPS;
```

All Rows Fetched: 18 in 0,004 seconds

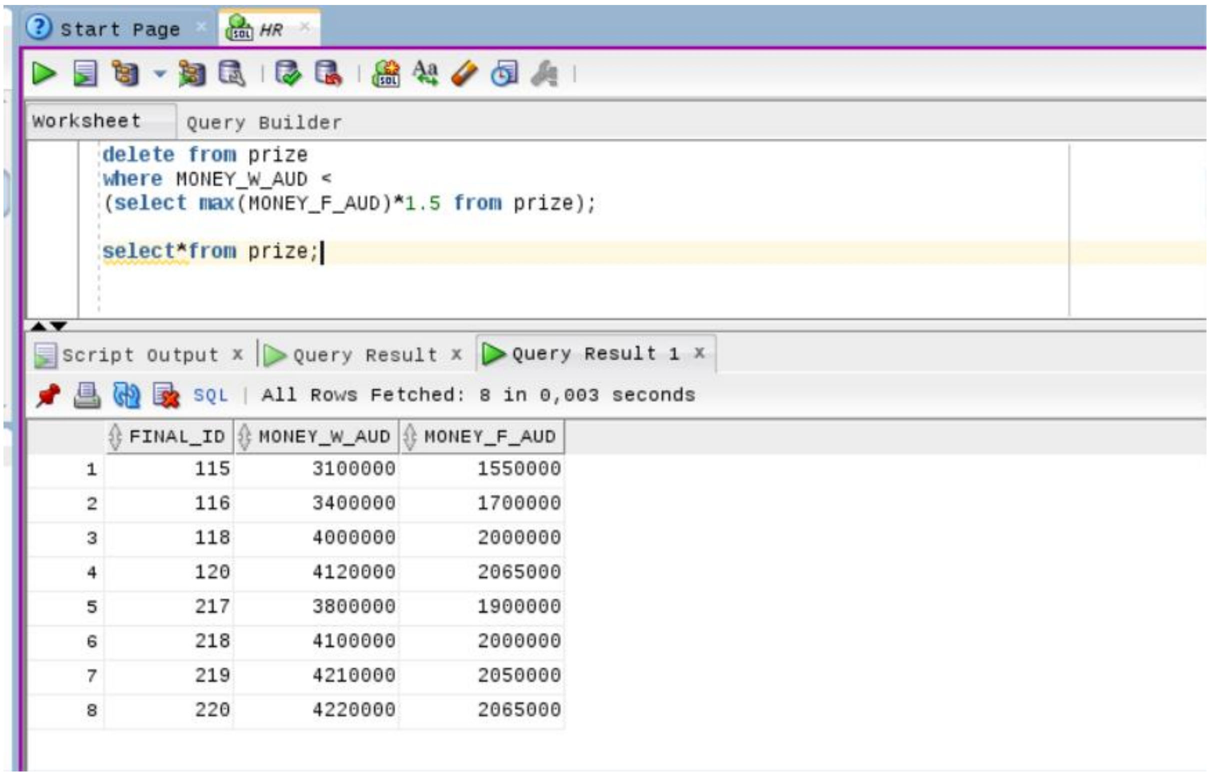
	RUNNER_UP_ID	NAME	COUNTRY	SEX
1	AR_CL	Arnaud Clement	France	M
2	RA_SC	Rainer Schüttler	Germany	M
3	LL_HE	Lleyton Hewitt	Germany	M
4	MA_BA	Marcos Baghdatis	Cyprus	M
5	FE_GO	Fernando Gonzalez	Chile	M
6	JO_TS	Jo-Wilfried Tsonga	France	M
7	AN_MU	Andy Murray	United Kingdom	M
8	MA_CI	Marin Cilic	Croatia	M
9	DO_TH	Dominic Thiem	Croatia	M
10	MA_HI	Martina Hingis	Switzerland	F
11	VE_WI	Venus Williams	USA	F
12	LI_DA	Lindsay Davenport	USA	F
13	AN_IV	Ana Ivanovic	Serbia	F
14	DI_SA	Dinara Safina	Russia	F
15	DO_CI	Dominika Cibulkova	Slovakia	F
16	SI_HA	Simona Halep	Romania	F
17	PE_KV	Petra Kvitova	Czeh Republic	F
18	GA_MU	Garbine Muguruza	Spain	F

- **Remove entries from the Prize table** where the first-place prize is less than 1.5 times the second-place prize.



```
delete from prize
where MONEY_W_AUD <
(select max(MONEY_F_AUD)*1.5 from prize);
```

```
Task completed in 0,002 seconds
0 rows deleted.

4 rows deleted.
```

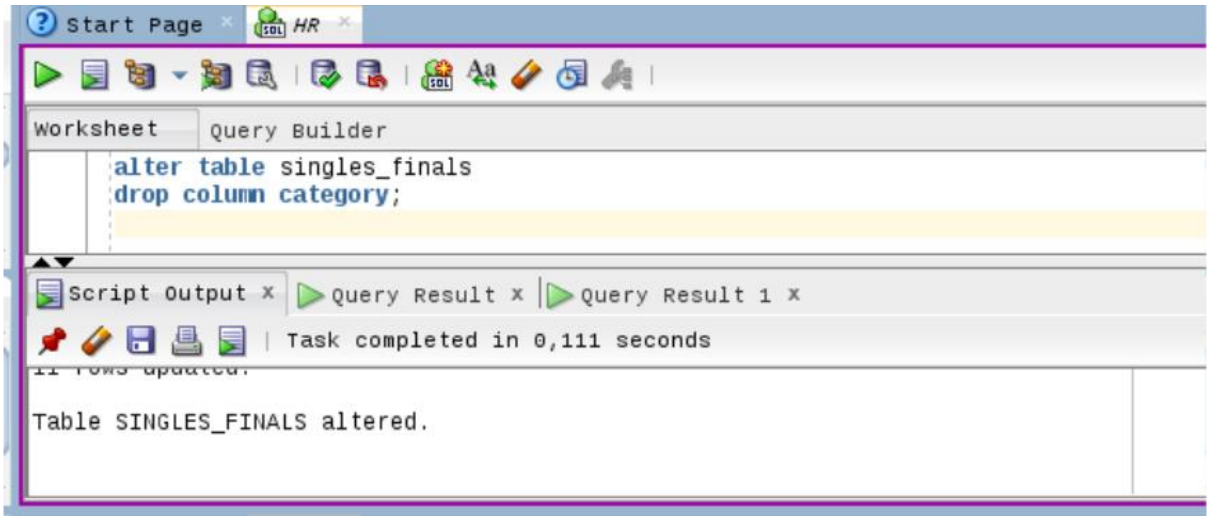


```
delete from prize
where MONEY_W_AUD <
(select max(MONEY_F_AUD)*1.5 from prize);

select*from prize;
```

All Rows Fetched: 8 in 0,003 seconds

	FINAL_ID	MONEY_W_AUD	MONEY_F_AUD
1	115	3100000	1550000
2	116	3400000	1700000
3	118	4000000	2000000
4	120	4120000	2065000
5	217	3800000	1900000
6	218	4100000	2000000
7	219	4210000	2050000
8	220	4220000	2065000

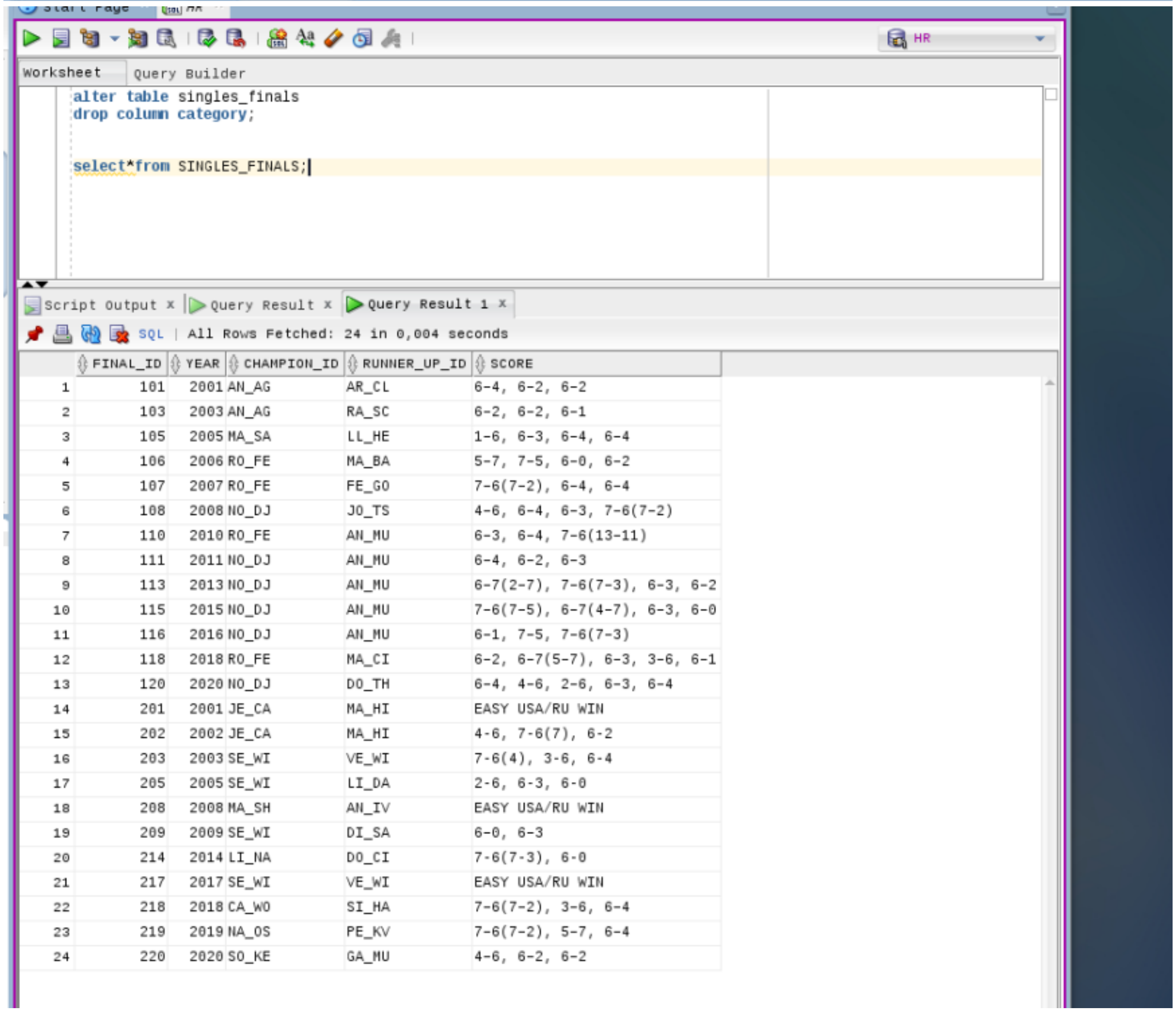
- **Delete the "Category" column** from the Singles Finals table.



```
alter table singles_finals
drop column category;
```

Task completed in 0,111 seconds

```
Table SINGLES_FINALS altered.
```

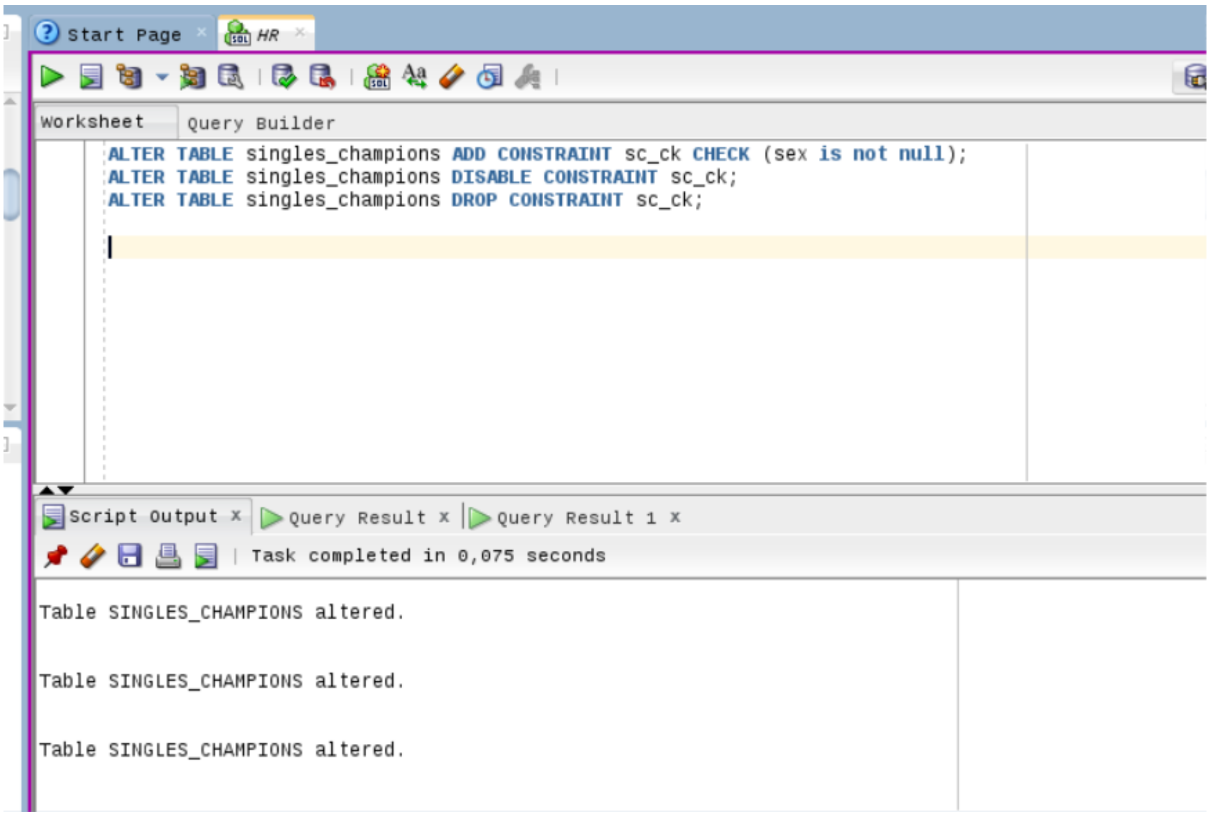


```
alter table singles_finals
drop column category;

select*from SINGLES_FINALS;
```

All Rows Fetched: 24 in 0,004 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	SCORE
1	101	2001	AN_AG	AR_CL	6-4, 6-2, 6-2
2	103	2003	AN_AG	RA_SC	6-2, 6-2, 6-1
3	105	2005	MA_SA	LL_HE	1-6, 6-3, 6-4, 6-4
4	106	2006	RO_FE	MA_BA	5-7, 7-5, 6-0, 6-2
5	107	2007	RO_FE	FE_GO	7-6(7-2), 6-4, 6-4
6	108	2008	NO_DJ	JO_TS	4-6, 6-4, 6-3, 7-6(7-2)
7	110	2010	RO_FE	AN_MU	6-3, 6-4, 7-6(13-11)
8	111	2011	NO_DJ	AN_MU	6-4, 6-2, 6-3
9	113	2013	NO_DJ	AN_MU	6-7(2-7), 7-6(7-3), 6-3, 6-2
10	115	2015	NO_DJ	AN_MU	7-6(7-5), 6-7(4-7), 6-3, 6-0
11	116	2016	NO_DJ	AN_MU	6-1, 7-5, 7-6(7-3)
12	118	2018	RO_FE	MA_CI	6-2, 6-7(5-7), 6-3, 3-6, 6-1
13	120	2020	NO_DJ	DO_TH	6-4, 4-6, 2-6, 6-3, 6-4
14	201	2001	JE_CA	MA_HI	EASY USA/RU WIN
15	202	2002	JE_CA	MA_HI	4-6, 7-6(7), 6-2
16	203	2003	SE_WI	VE_WI	7-6(4), 3-6, 6-4
17	205	2005	SE_WI	LI_DA	2-6, 6-3, 6-0
18	208	2008	MA_SH	AN_IV	EASY USA/RU WIN
19	209	2009	SE_WI	DI_SA	6-0, 6-3
20	214	2014	LI_NA	DO_CI	7-6(7-3), 6-0
21	217	2017	SE_WI	VE_WI	EASY USA/RU WIN
22	218	2018	CA_WO	SI_HA	7-6(7-2), 3-6, 6-4
23	219	2019	NA_OS	PE_KV	7-6(7-2), 5-7, 6-4
24	220	2020	SO_KE	GA_MU	4-6, 6-2, 6-2

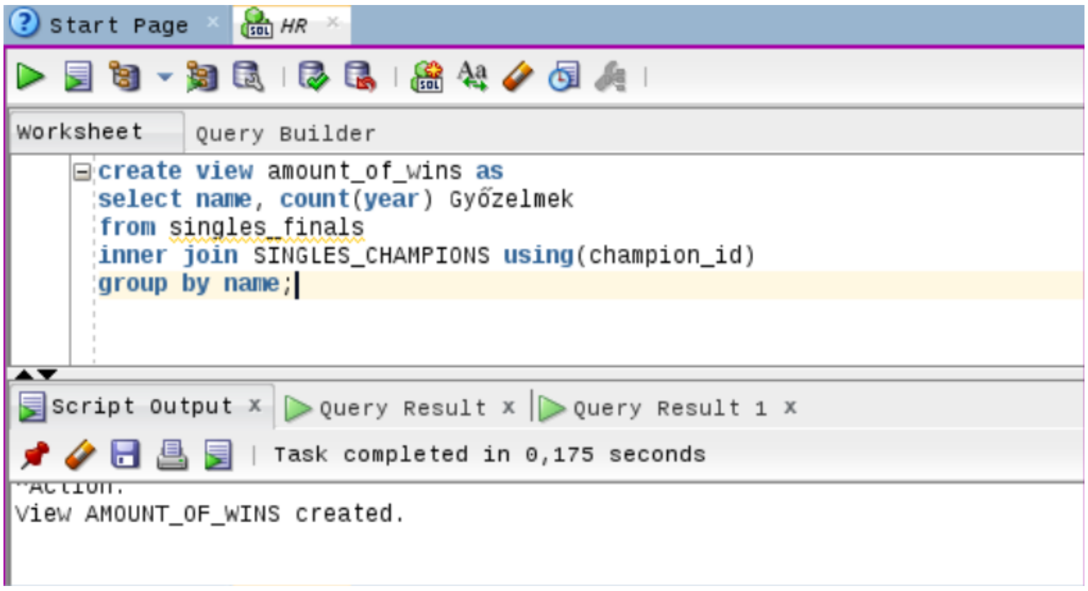
- **Add, restrict, and remove the "Sex is not null" constraint** in respective steps.



```
ALTER TABLE singles_champions ADD CONSTRAINT sc_ck CHECK (sex is not null);
ALTER TABLE singles_champions DISABLE CONSTRAINT sc_ck;
ALTER TABLE singles_champions DROP CONSTRAINT sc_ck;
```

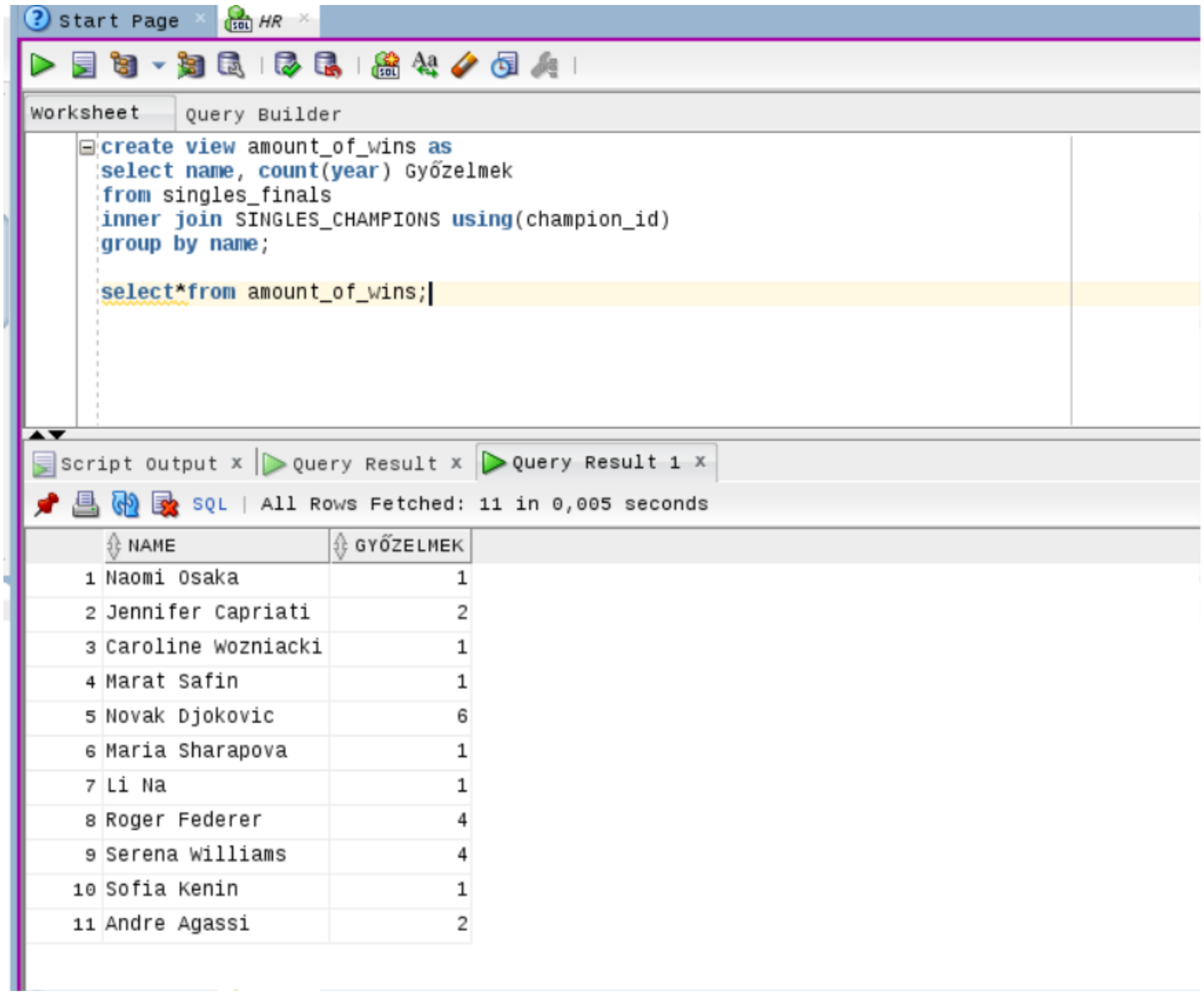
```
Task completed in 0,075 seconds

Table SINGLES_CHAMPIONS altered.


Table SINGLES_CHAMPIONS altered.


Table SINGLES_CHAMPIONS altered.
```

- **Create views**:
    - A view showing players' names and their win counts.



```
create view amount_of_wins as
select name, count(year) Győzelmek
from singles_finals
inner join SINGLES_CHAMPIONS using(champion_id)
group by name;
```

```
Task completed in 0,175 seconds
*Action:
View AMOUNT_OF_WINS created.
```

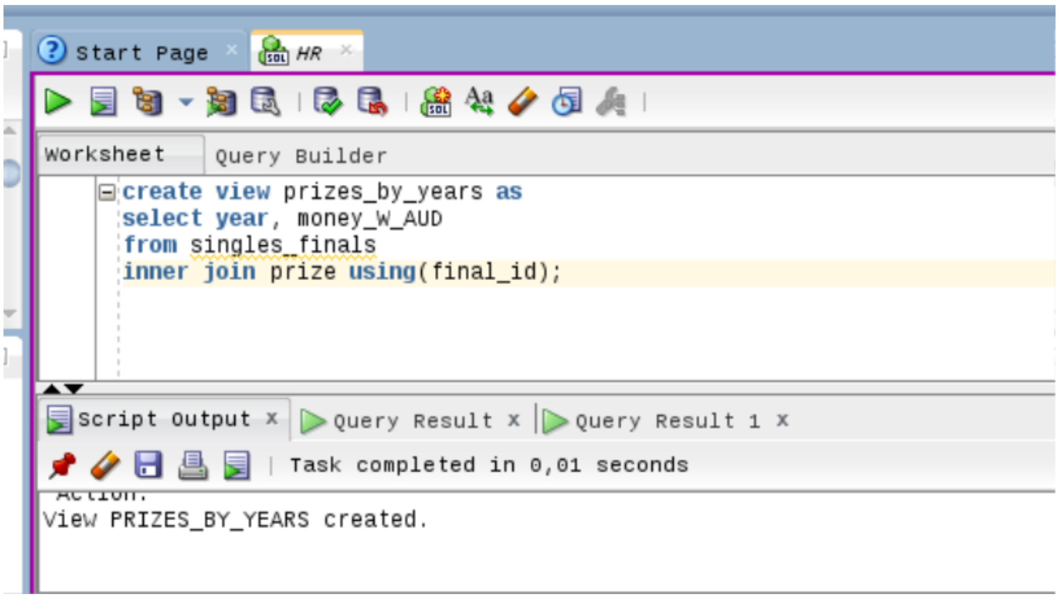


```
create view amount_of_wins as
select name, count(year) Győzelmek
from singles_finals
inner join SINGLES_CHAMPIONS using(champion_id)
group by name;

select*from amount_of_wins;
```

All Rows Fetched: 11 in 0,005 seconds

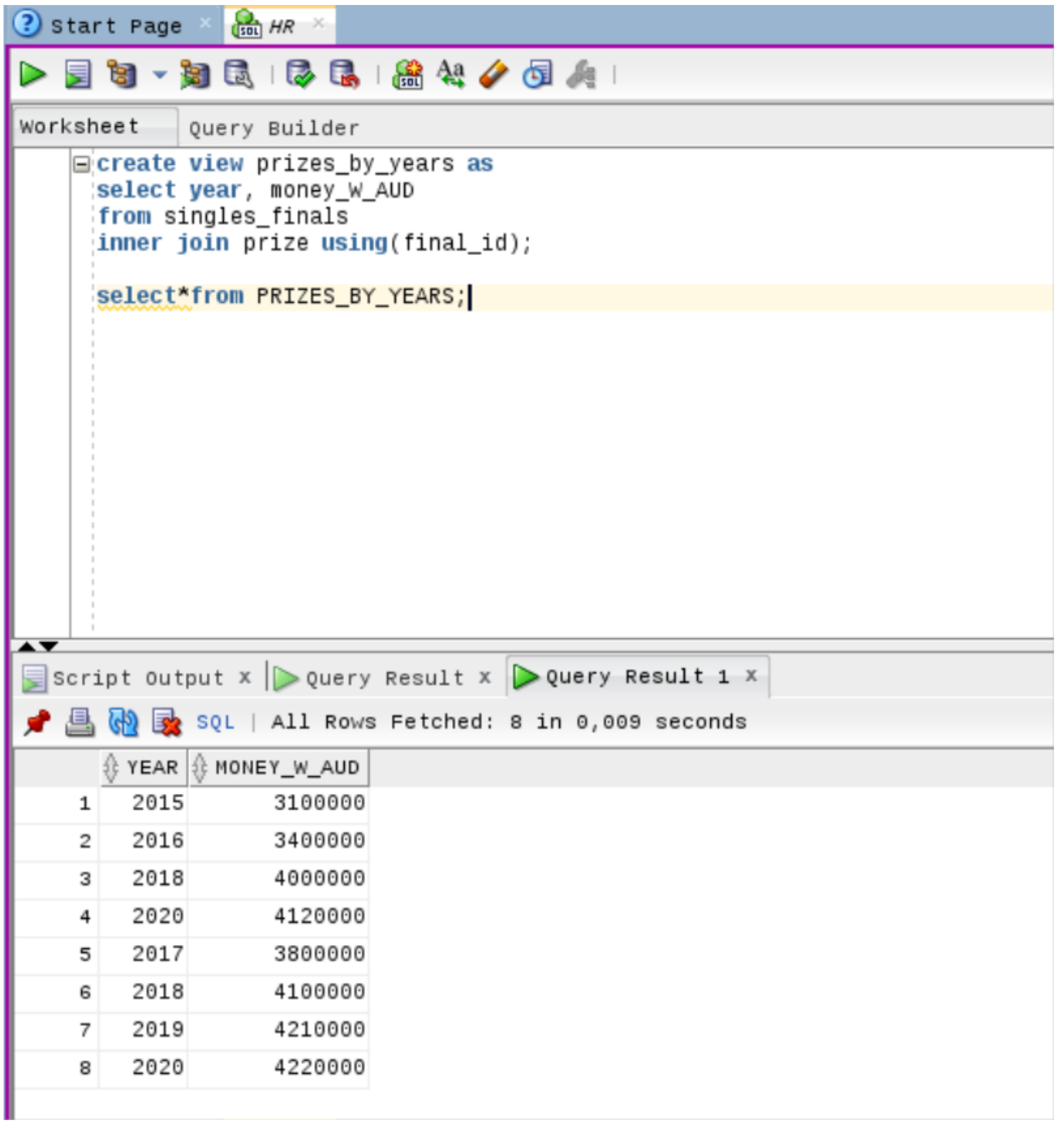
	NAME	GYŐZELMEK
1	Naomi Osaka	1
2	Jennifer Capriati	2
3	Caroline Wozniacki	1
4	Marat Safin	1
5	Novak Djokovic	6
6	Maria Sharapova	1
7	Li Na	1
8	Roger Federer	4
9	Serena Williams	4
10	Sofia Kenin	1
11	Andre Agassi	2

    - A view showing prizes categorized by year.



```
create view prizes_by_years as
select year, money_W_AUD
from singles_finals
inner join prize using(final_id);
```

```
Task completed in 0,01 seconds
Action:
View PRIZES_BY_YEARS created.
```

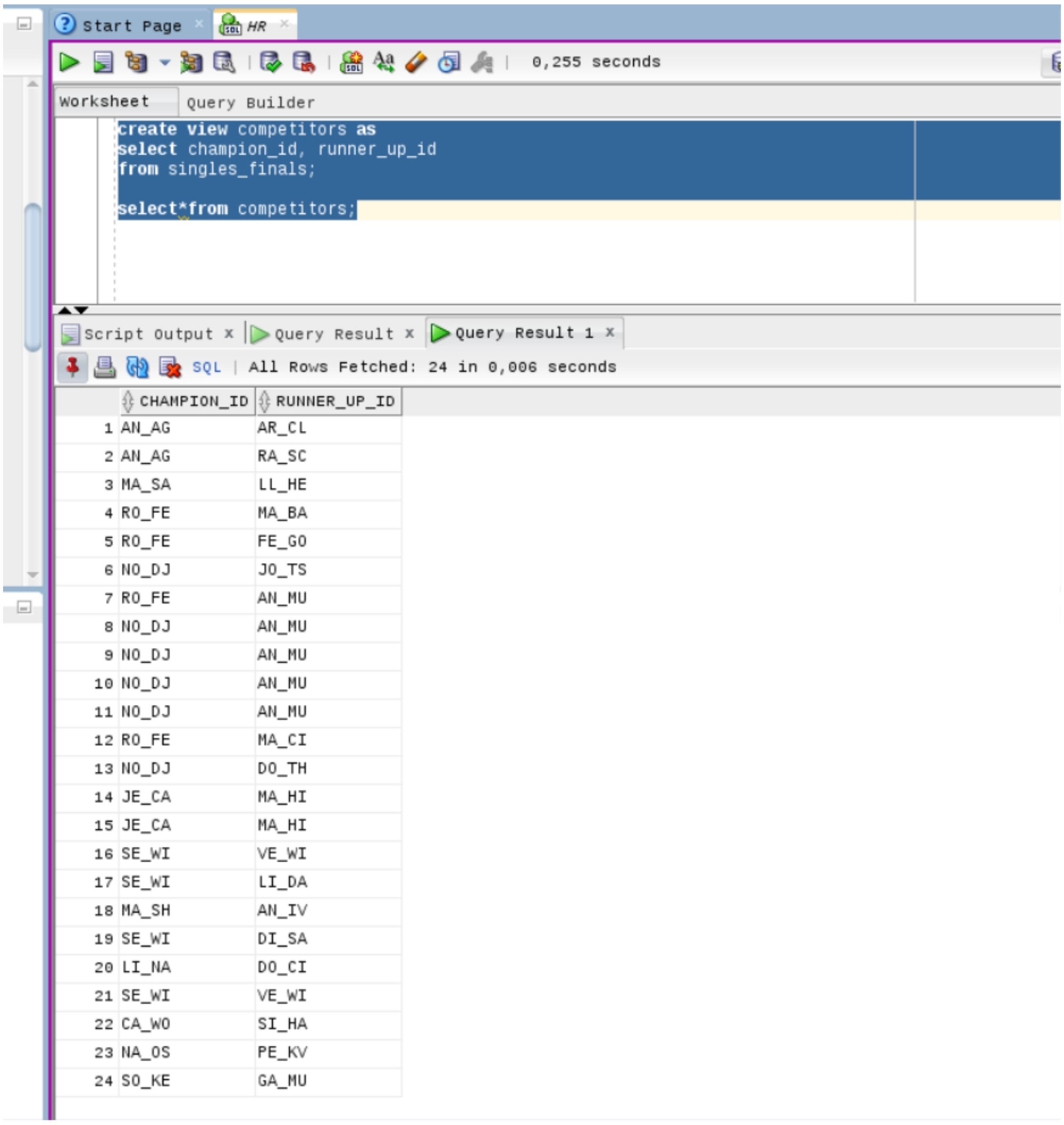


```
create view prizes_by_years as
select year, money_W_AUD
from singles_finals
inner join prize using(final_id);

select*from PRIZES_BY_YEARS;
```

All Rows Fetched: 8 in 0,009 seconds

	YEAR	MONEY_W_AUD
1	2015	3100000
2	2016	3400000
3	2018	4000000
4	2020	4120000
5	2017	3800000
6	2018	4100000
7	2019	4210000
8	2020	4220000

- A view displaying IDs of competing players.

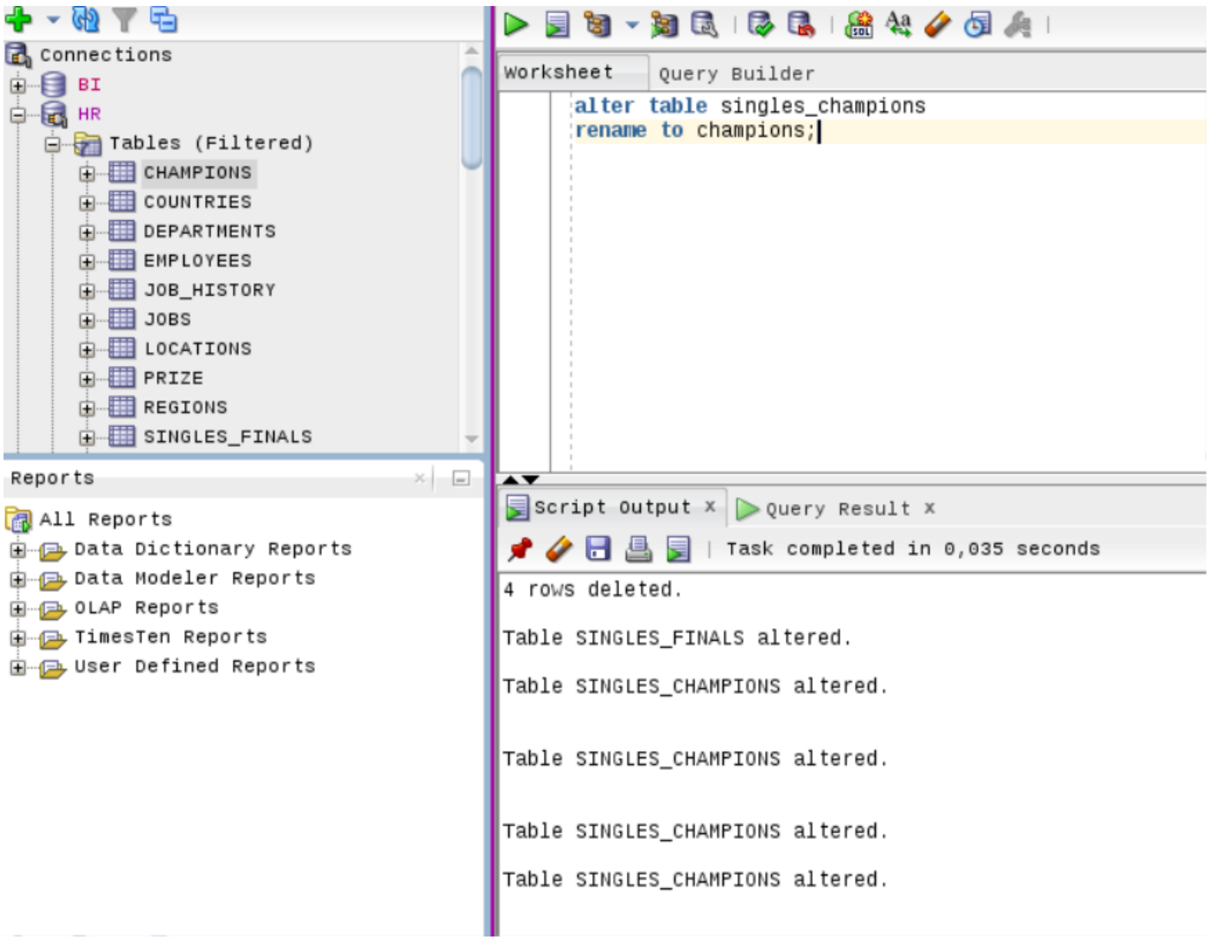


```
create view competitors as
select champion_id, runner_up_id
from singles_finals;

select*from competitors;
```

All Rows Fetched: 24 in 0,006 seconds

	CHAMPION_ID	RUNNER_UP_ID
1	AN_AG	AR_CL
2	AN_AG	RA_SC
3	MA_SA	LL_HE
4	RO_FE	MA_BA
5	RO_FE	FE_GO
6	NO_DJ	JO_TS
7	RO_FE	AN_MU
8	NO_DJ	AN_MU
9	NO_DJ	AN_MU
10	NO_DJ	AN_MU
11	NO_DJ	AN_MU
12	RO_FE	MA_CI
13	NO_DJ	DO_TH
14	JE_CA	MA_HI
15	JE_CA	MA_HI
16	SE_WI	VE_WI
17	SE_WI	LI_DA
18	MA_SH	AN_IV
19	SE_WI	DI_SA
20	LI_NA	DO_CI
21	SE_WI	VE_WI
22	CA_WO	SI_HA
23	NA_OS	PE_KV
24	SO_KE	GA_MU

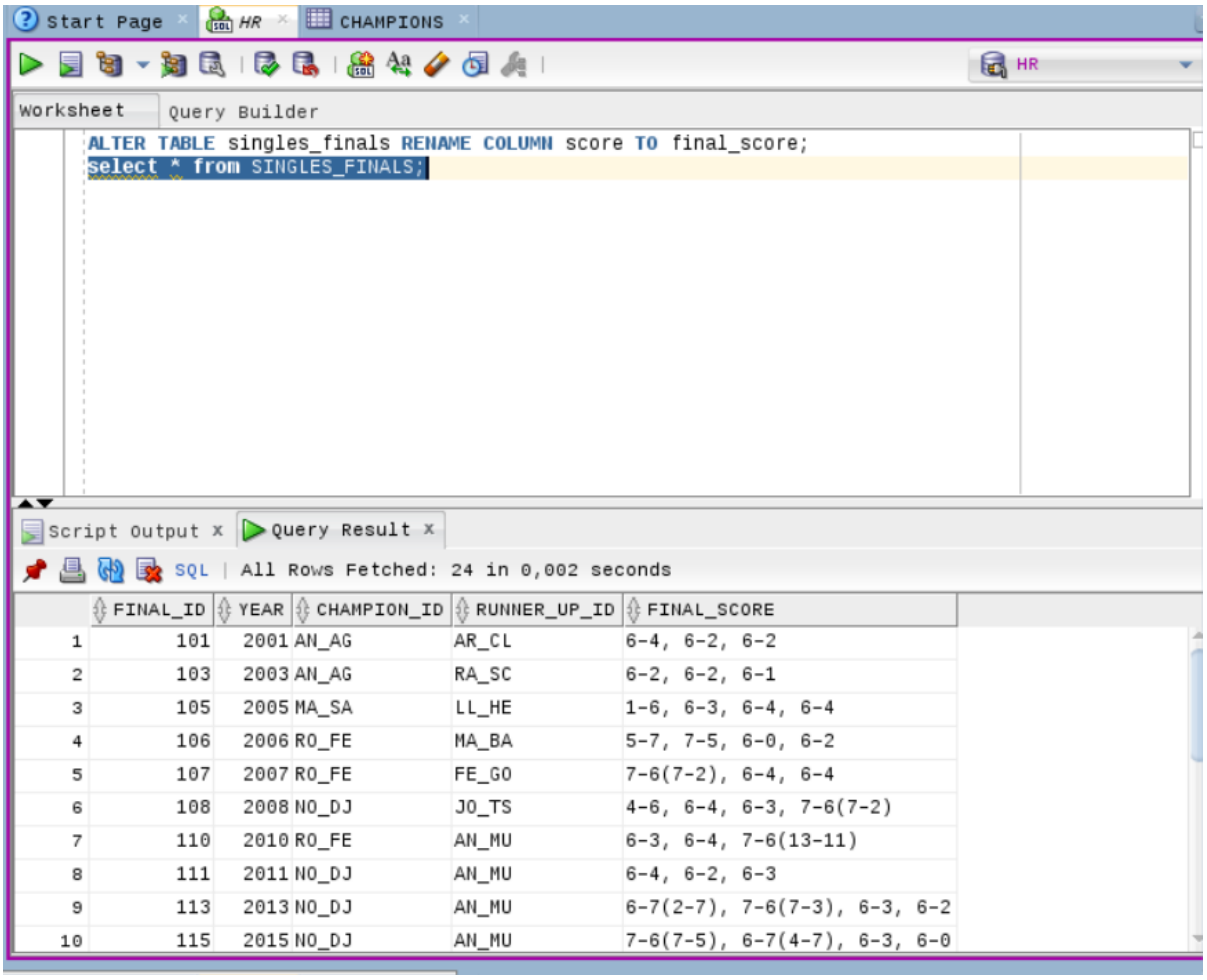
- **Rename tables and columns**:
  - Rename the *Singles Champions* table to *Champions*.



```
alter table singles_champions
rename to champions;
```

```
4 rows deleted.

Table SINGLES_FINALS altered.

Table SINGLES_CHAMPIONS altered.


Table SINGLES_CHAMPIONS altered.


Table SINGLES_CHAMPIONS altered.

Table SINGLES_CHAMPIONS altered.
```

- Rename the *Score* column in *Singles Finals* to *Final_Score*.



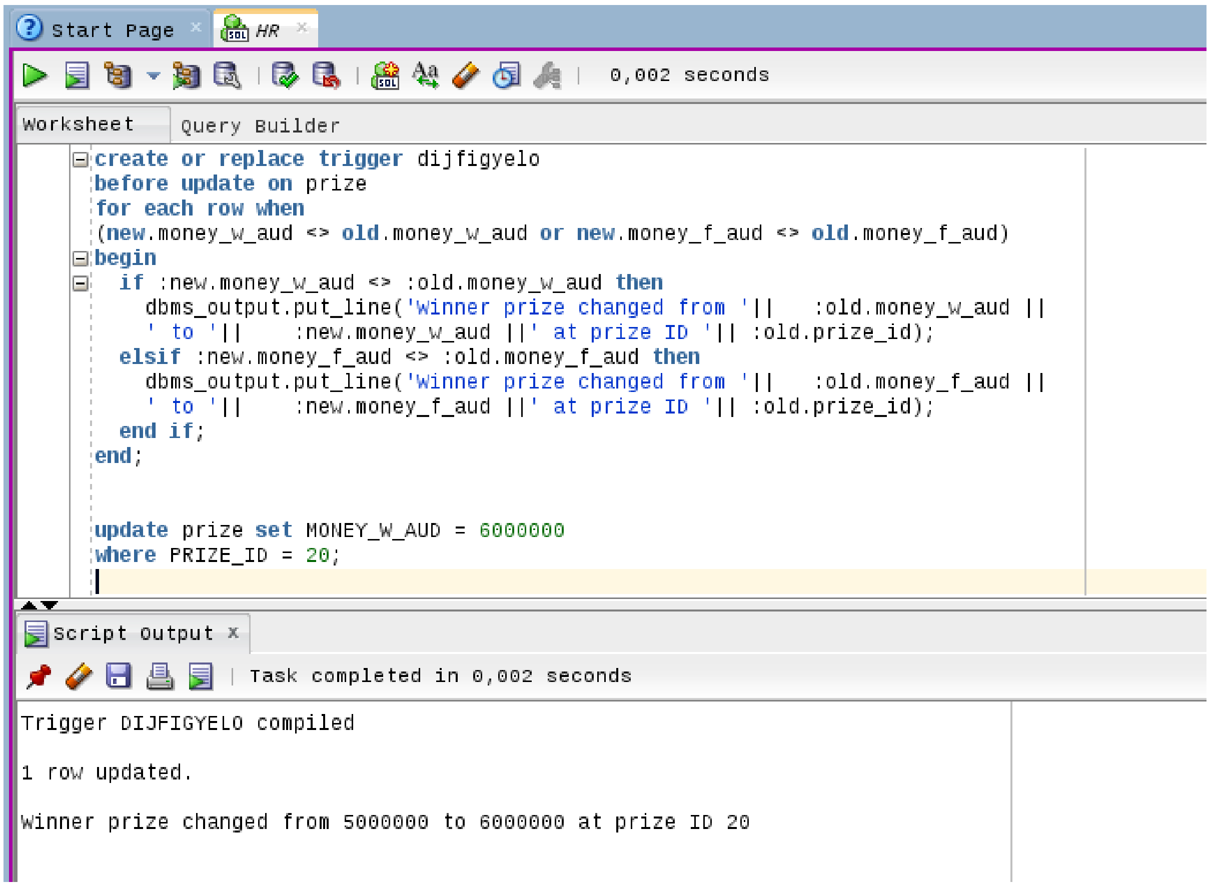
```
ALTER TABLE singles_finals RENAME COLUMN score TO final_score;
select * from SINGLES_FINALS;
```

All Rows Fetched: 24 in 0,002 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	FINAL_SCORE
1	101	2001	AN_AG	AR_CL	6-4, 6-2, 6-2
2	103	2003	AN_AG	RA_SC	6-2, 6-2, 6-1
3	105	2005	MA_SA	LL_HE	1-6, 6-3, 6-4, 6-4
4	106	2006	RO_FE	MA_BA	5-7, 7-5, 6-0, 6-2
5	107	2007	RO_FE	FE_GO	7-6(7-2), 6-4, 6-4
6	108	2008	NO_DJ	JO_TS	4-6, 6-4, 6-3, 7-6(7-2)
7	110	2010	RO_FE	AN_MU	6-3, 6-4, 7-6(13-11)
8	111	2011	NO_DJ	AN_MU	6-4, 6-2, 6-3
9	113	2013	NO_DJ	AN_MU	6-7(2-7), 7-6(7-3), 6-3, 6-2
10	115	2015	NO_DJ	AN_MU	7-6(7-5), 6-7(4-7), 6-3, 6-0

11. PL/SQL, Transaction, and Permission Management

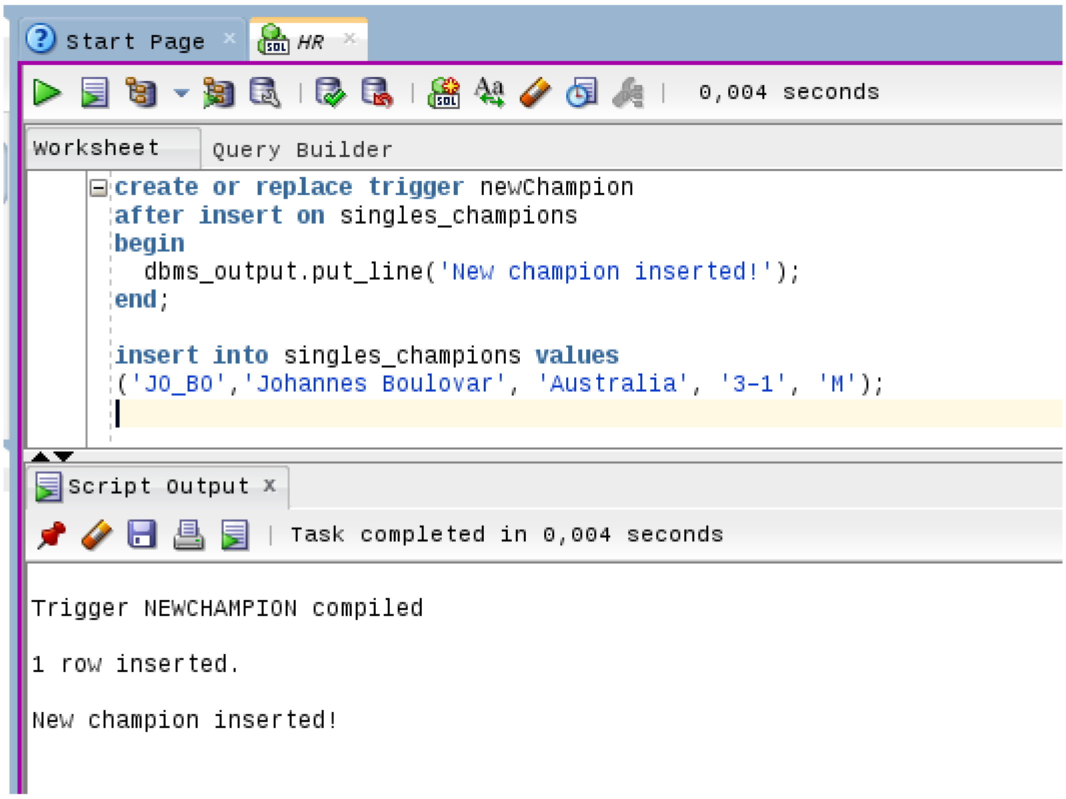
- **When modifying prize values** in the Prize table, **log** the original and new amounts along with the Prize ID.



```
create or replace trigger dijfigyelo
before update on prize
for each row when
(new.money_w_aud <> old.money_w_aud or new.money_f_aud <> old.money_f_aud)
begin
  if :new.money_w_aud <> :old.money_w_aud then
    dbms_output.put_line('Winner prize changed from '||   :old.money_w_aud ||
    ' to '||    :new.money_w_aud ||' at prize ID '|| :old.prize_id);
  elsif :new.money_f_aud <> :old.money_f_aud then
    dbms_output.put_line('Winner prize changed from '||   :old.money_f_aud ||
    ' to '||    :new.money_f_aud ||' at prize ID '|| :old.prize_id);
  end if;
end;


update prize set MONEY_W_AUD = 6000000
where PRIZE_ID = 20;
```

```
Trigger DIJFIGYELO compiled

1 row updated.

Winner prize changed from 5000000 to 6000000 at prize ID 20
```

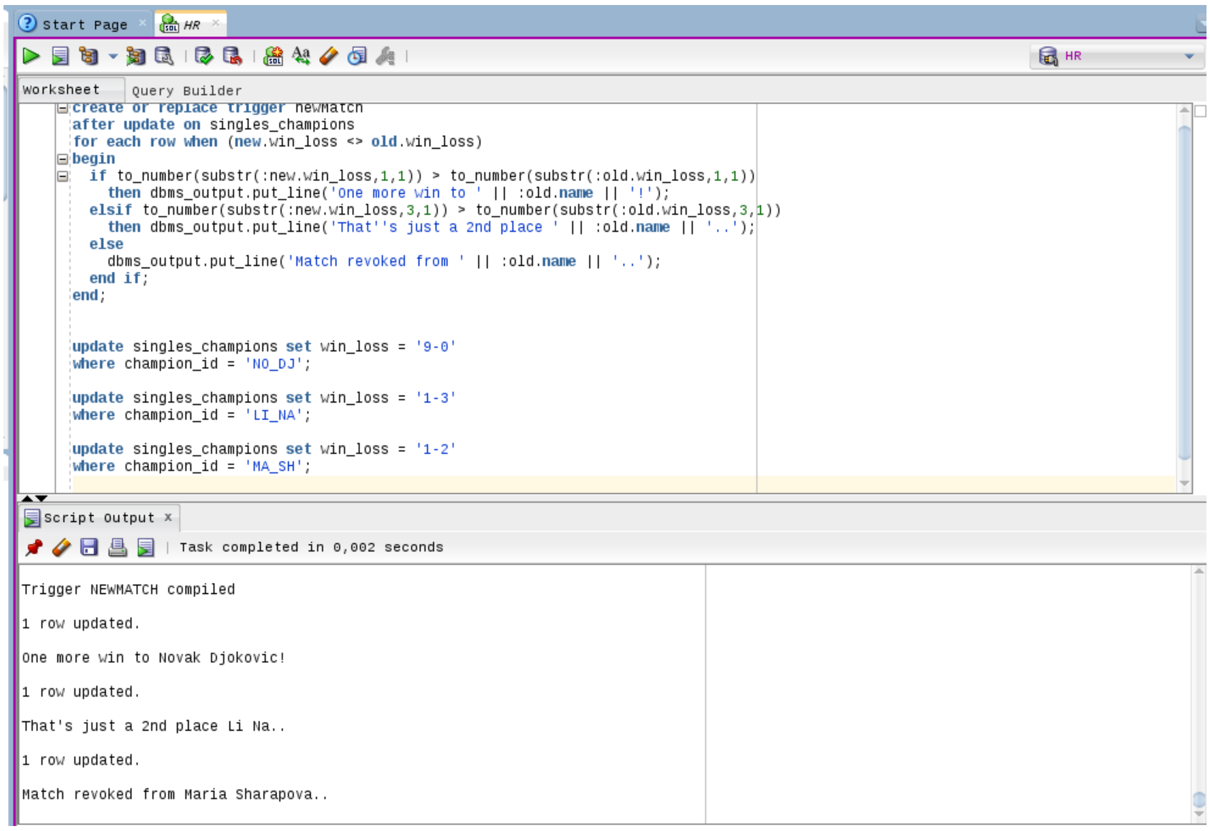
- **On inserting a new row** into the Singles Champions table, **log** "New champion inserted!"



```
create or replace trigger newChampion
after insert on singles_champions
begin
  dbms_output.put_line('New champion inserted!');
end;

insert into singles_champions values
('JO_BO','Johannes Boulovar', 'Australia', '3-1', 'M');
```

```
Trigger NEWCHAMPION compiled

1 row inserted.

New champion inserted!
```

- **Track changes in the Win-Loss field** in the Singles Champions table:
    - Log "One more win to [player name]!" if the win count increases.
    - Log "That's just a 2nd place [player name].." if the loss-count increases.
    - Log "Match revoked from [player name].." if neither condition is met, implying a match was revoked.

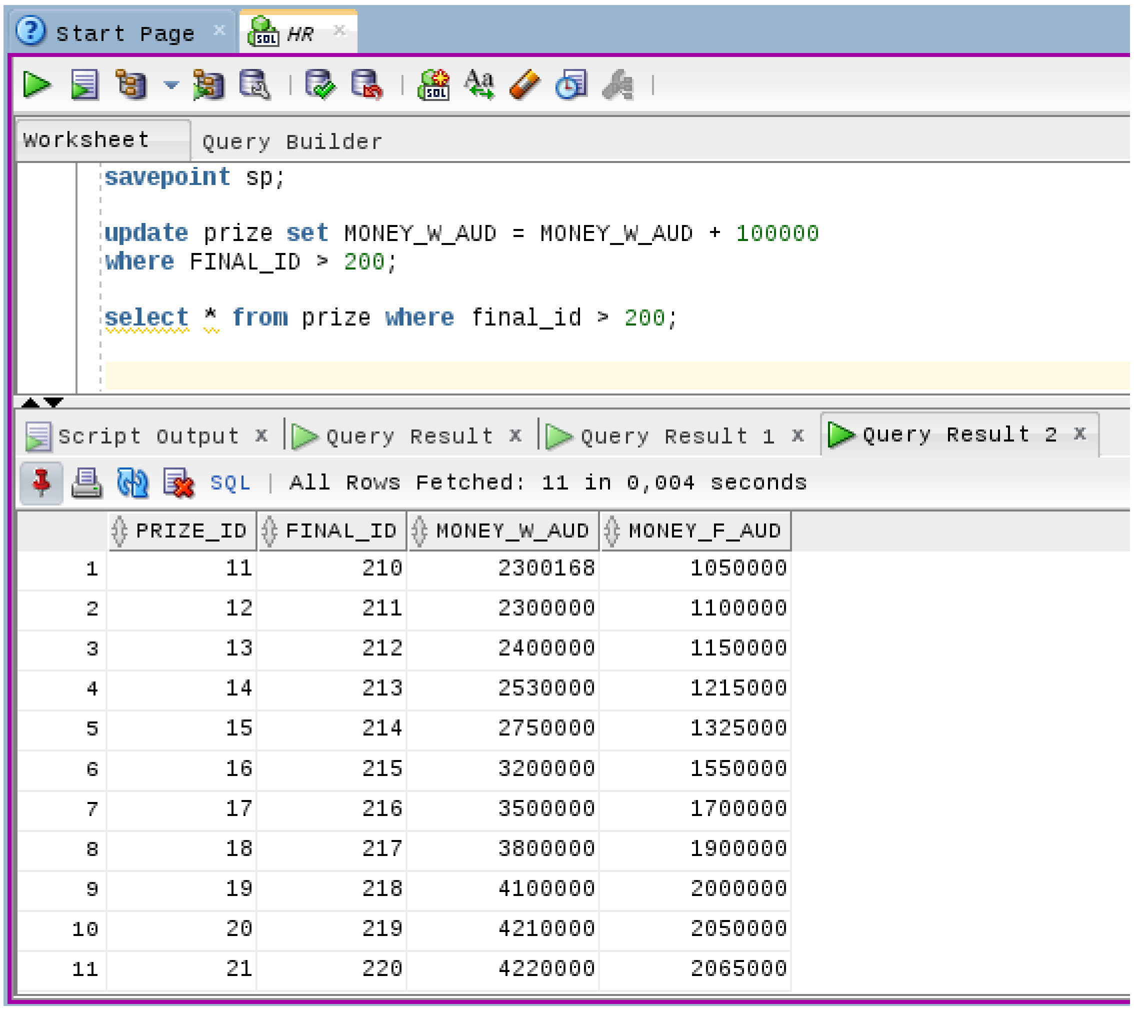


```
create or replace trigger newMatch
after update on singles_champions
for each row when (new.win_loss <> old.win_loss)
begin
  if to_number(substr(:new.win_loss,1,1)) > to_number(substr(:old.win_loss,1,1))
    then dbms_output.put_line('One more win to ' || :old.name || '!');
  elsif to_number(substr(:new.win_loss,3,1)) > to_number(substr(:old.win_loss,3,1))
    then dbms_output.put_line('That''s just a 2nd place ' || :old.name || '..');
  else
    dbms_output.put_line('Match revoked from ' || :old.name || '..');
  end if;
end;


update singles_champions set win_loss = '9-0'
where champion_id = 'NO_DJ';

update singles_champions set win_loss = '1-3'
where champion_id = 'LI_NA';

update singles_champions set win_loss = '1-2'
where champion_id = 'MA_SH';
```

Task completed in 0,002 seconds

```
Trigger NEWMATCH compiled

1 row updated.

One more win to Novak Djokovic!

1 row updated.

That's just a 2nd place Li Na..

1 row updated.

Match revoked from Maria Sharapova..
```

- **Create a savepoint,** increase prizes for winners of finals with Final ID > 200 by AUD 100,000, rollback, and later commit the changes.

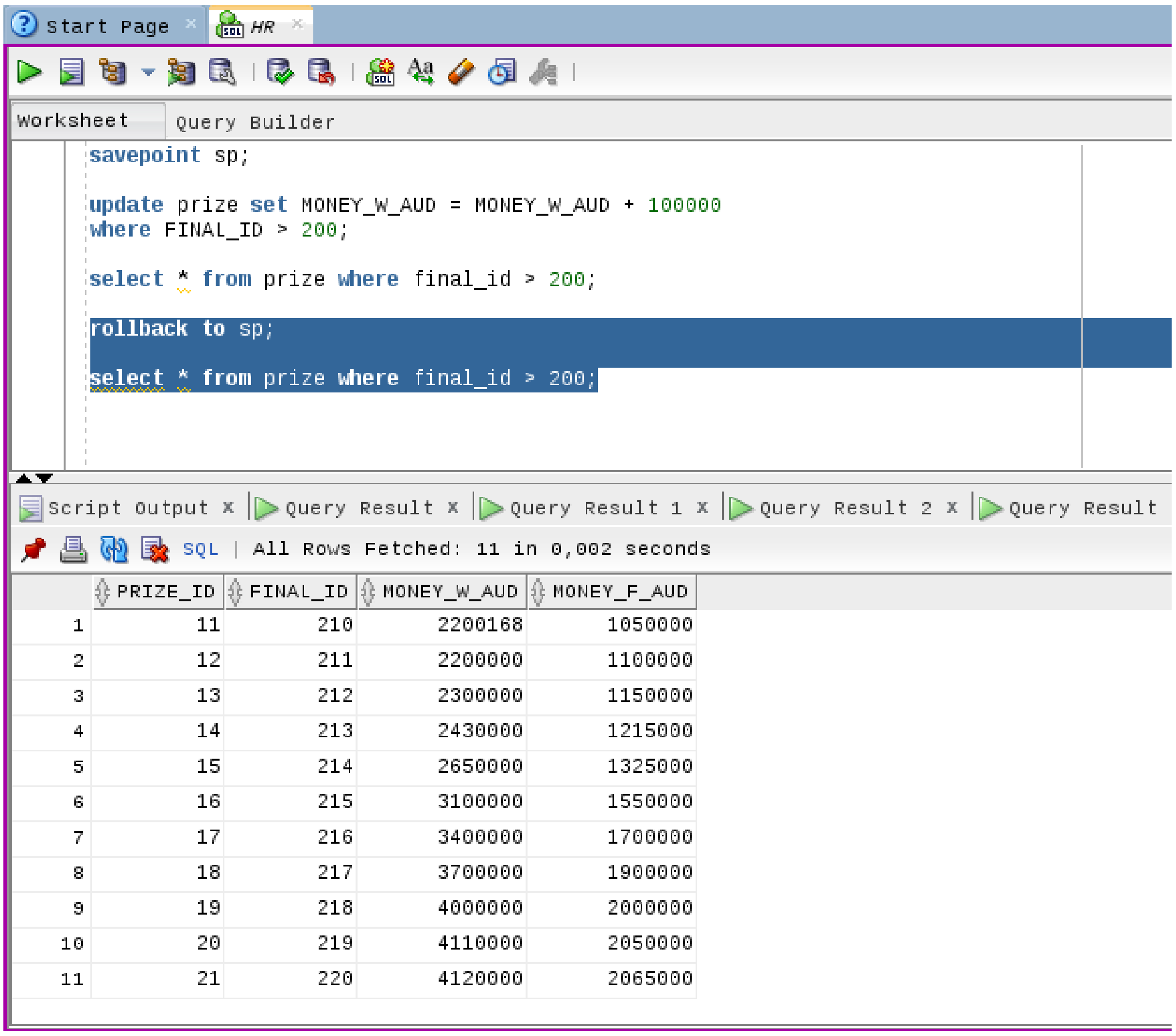


```
savepoint sp;

update prize set MONEY_W_AUD = MONEY_W_AUD + 100000
where FINAL_ID > 200;

select * from prize where final_id > 200;
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 11 in 0,004 seconds

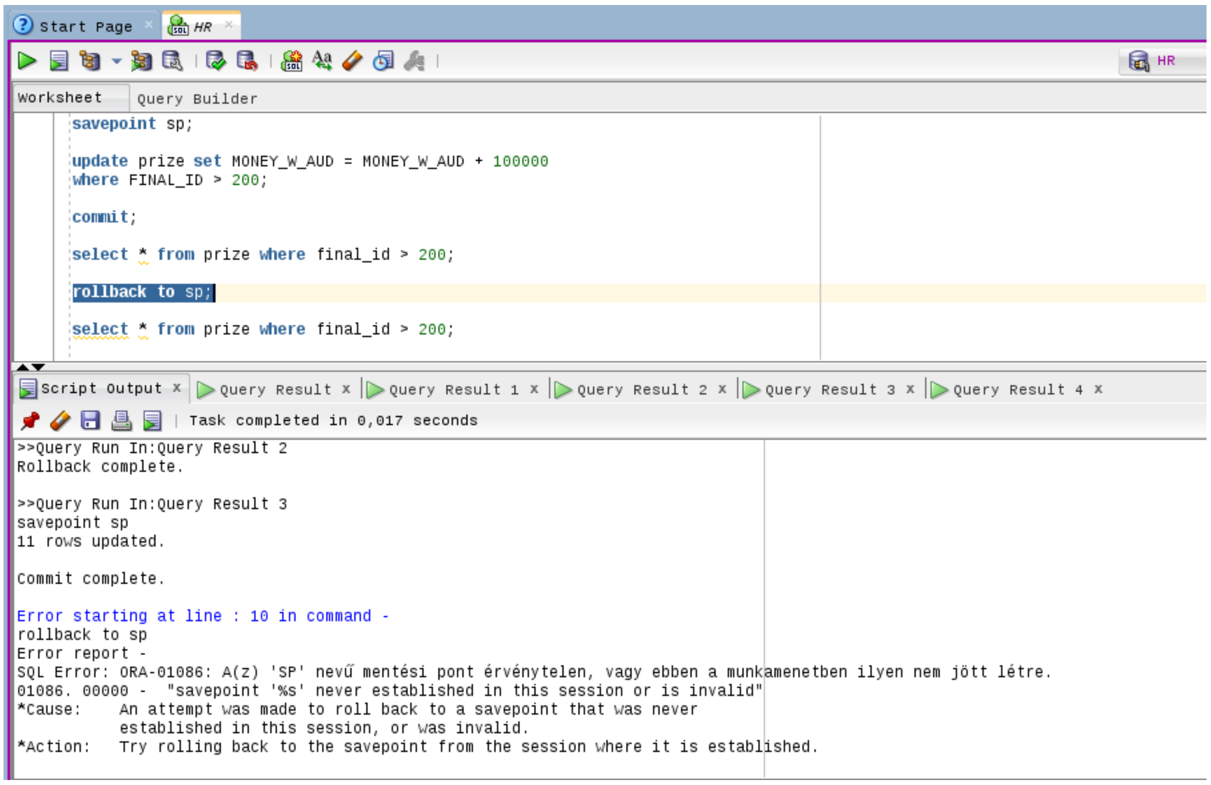
	PRIZE_ID	FINAL_ID	MONEY_W_AUD	MONEY_F_AUD
1	11	210	2300168	1050000
2	12	211	2300000	1100000
3	13	212	2400000	1150000
4	14	213	2530000	1215000
5	15	214	2750000	1325000
6	16	215	3200000	1550000
7	17	216	3500000	1700000
8	18	217	3800000	1900000
9	19	218	4100000	2000000
10	20	219	4210000	2050000
11	21	220	4220000	2065000



```
savepoint sp;

update prize set MONEY_W_AUD = MONEY_W_AUD + 100000
where FINAL_ID > 200;

select * from prize where final_id > 200;

rollback to sp;

select * from prize where final_id > 200;
```

All Rows Fetched: 11 in 0,002 seconds

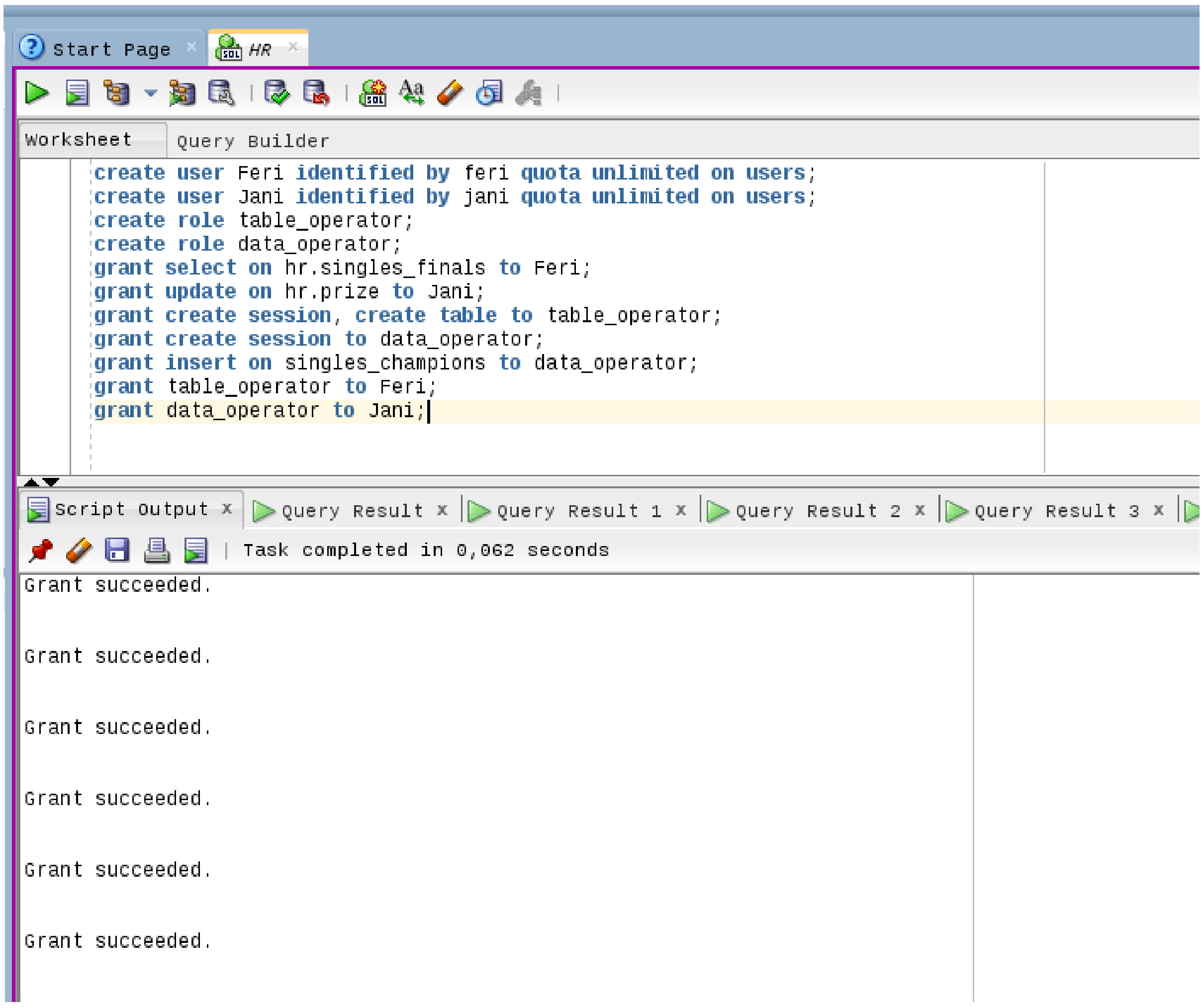
	PRIZE_ID	FINAL_ID	MONEY_W_AUD	MONEY_F_AUD
1	11	210	2200168	1050000
2	12	211	2200000	1100000
3	13	212	2300000	1150000
4	14	213	2430000	1215000
5	15	214	2650000	1325000
6	16	215	3100000	1550000
7	17	216	3400000	1700000
8	18	217	3700000	1900000
9	19	218	4000000	2000000
10	20	219	4110000	2050000
11	21	220	4120000	2065000



```
savepoint sp;

update prize set MONEY_W_AUD = MONEY_W_AUD + 100000
where FINAL_ID > 200;

commit;

select * from prize where final_id > 200;

rollback to sp;

select * from prize where final_id > 200;
```

```
>>Query Run In:Query Result 2
Rollback complete.

>>Query Run In:Query Result 3
savepoint sp
11 rows updated.

Commit complete.

Error starting at line : 10 in command -
rollback to sp
Error report -
SQL Error: ORA-01086: A(z) 'SP' nevű mentési pont érvénytelen, vagy ebben a munkamenetben ilyen nem jött létre.
01086. 00000 -  "savepoint '%s' never established in this session or is invalid"
*Cause:    An attempt was made to roll back to a savepoint that was never
           established in this session, or was invalid.
*Action:   Try rolling back to the savepoint from the session where it is established.
```

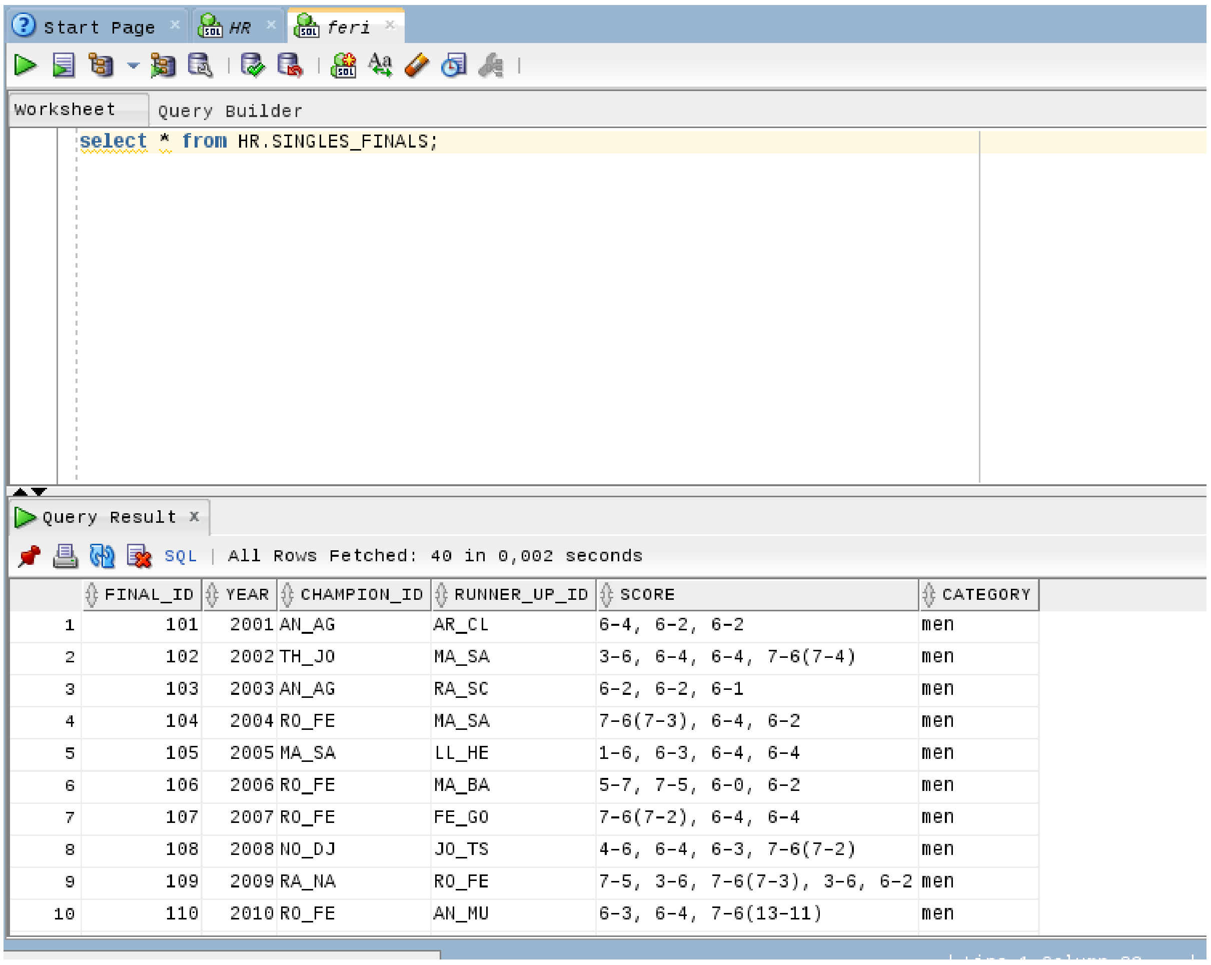
- **Set up user permissions**:
    - User *Feri* (password: *feri*), with the role of Table Operator, can list *Singles Finals* and create tables.
    - User *Jani* (password: *jani*), with the role of Data Operator, can modify the *Prize* table and insert new rows into the *Singles Champions* table.



```
create user Feri identified by feri quota unlimited on users;
create user Jani identified by jani quota unlimited on users;
create role table_operator;
create role data_operator;
grant select on hr.singles_finals to Feri;
grant update on hr.prize to Jani;
grant create session, create table to table_operator;
grant create session to data_operator;
grant insert on singles_champions to data_operator;
grant table_operator to Feri;
grant data_operator to Jani;
```

```
Task completed in 0,062 seconds
Grant succeeded.


Grant succeeded.


Grant succeeded.


Grant succeeded.


Grant succeeded.


Grant succeeded.
```

- **Testing**:
  - As *Feri*:
    - Successfully list *Singles Finals*.

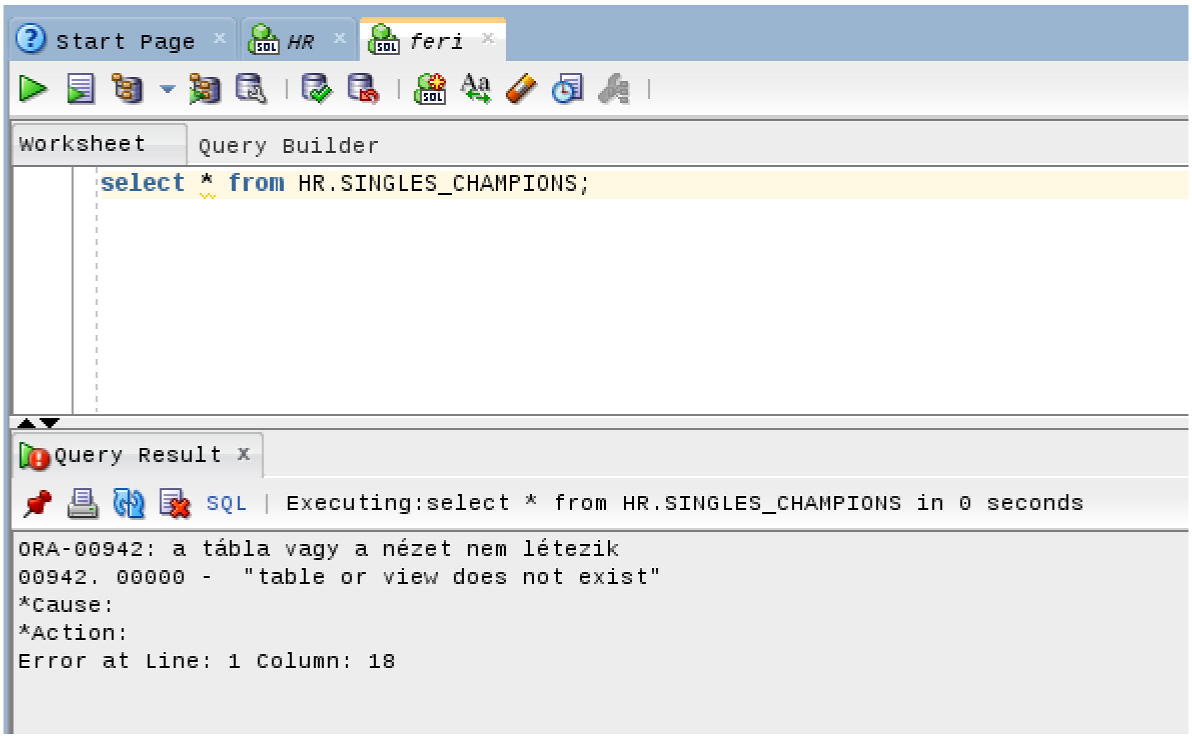


```
select * from HR.SINGLES_FINALS;
```

All Rows Fetched: 40 in 0,002 seconds

	FINAL_ID	YEAR	CHAMPION_ID	RUNNER_UP_ID	SCORE	CATEGORY
1	101	2001	AN_AG	AR_CL	6-4, 6-2, 6-2	men
2	102	2002	TH_JO	MA_SA	3-6, 6-4, 6-4, 7-6(7-4)	men
3	103	2003	AN_AG	RA_SC	6-2, 6-2, 6-1	men
4	104	2004	RO_FE	MA_SA	7-6(7-3), 6-4, 6-2	men
5	105	2005	MA_SA	LL_HE	1-6, 6-3, 6-4, 6-4	men
6	106	2006	RO_FE	MA_BA	5-7, 7-5, 6-0, 6-2	men
7	107	2007	RO_FE	FE_GO	7-6(7-2), 6-4, 6-4	men
8	108	2008	NO_DJ	JO_TS	4-6, 6-4, 6-3, 7-6(7-2)	men
9	109	2009	RA_NA	RO_FE	7-5, 3-6, 7-6(7-3), 3-6, 6-2	men
10	110	2010	RO_FE	AN_MU	6-3, 6-4, 7-6(13-11)	men

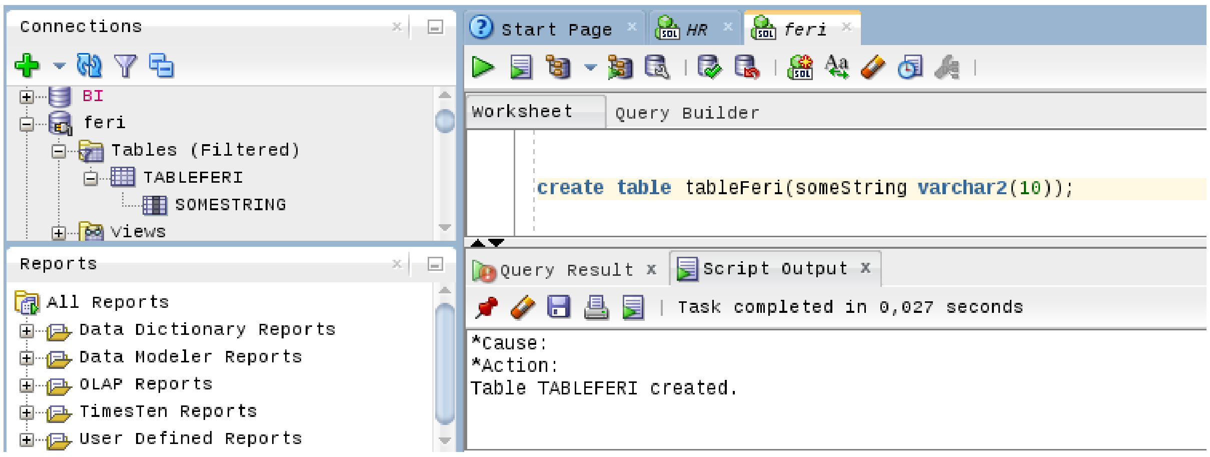
- Fail to list *Singles Champions*.



```
select * from HR.SINGLES_CHAMPIONS;
```

```
Executing:select * from HR.SINGLES_CHAMPIONS in 0 seconds
ORA-00942: a tábla vagy a nézet nem létezik
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
Error at Line: 1 Column: 18
```

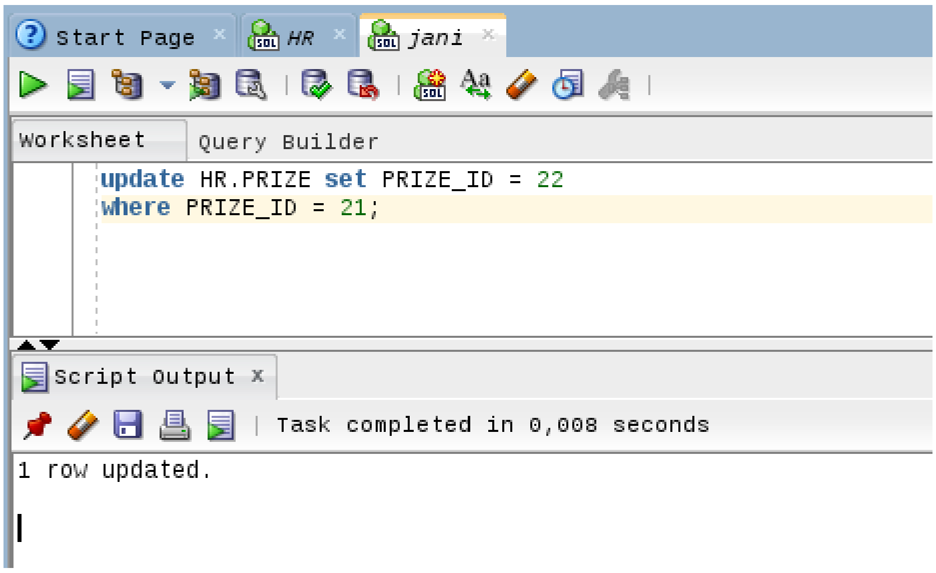
- Successfully create a table.



```
create table tableFeri(someString varchar2(10));
```

```
Task completed in 0,027 seconds
*Cause:
*Action:
Table TABLEFERI created.
```

- As *Jani*:
    - Successfully modify Prize ID 22 (from 21).

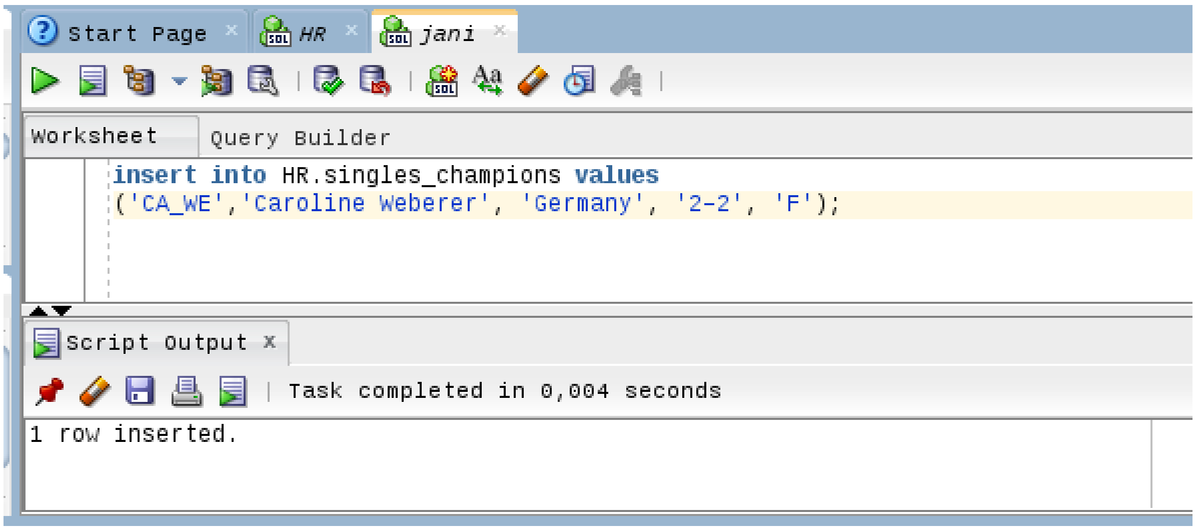


```
update HR.PRIZE set PRIZE_ID = 22
where PRIZE_ID = 21;
```

Task completed in 0,008 seconds

```
1 row updated.
```

    - Successfully insert a new row into the *Singles Champions* table.

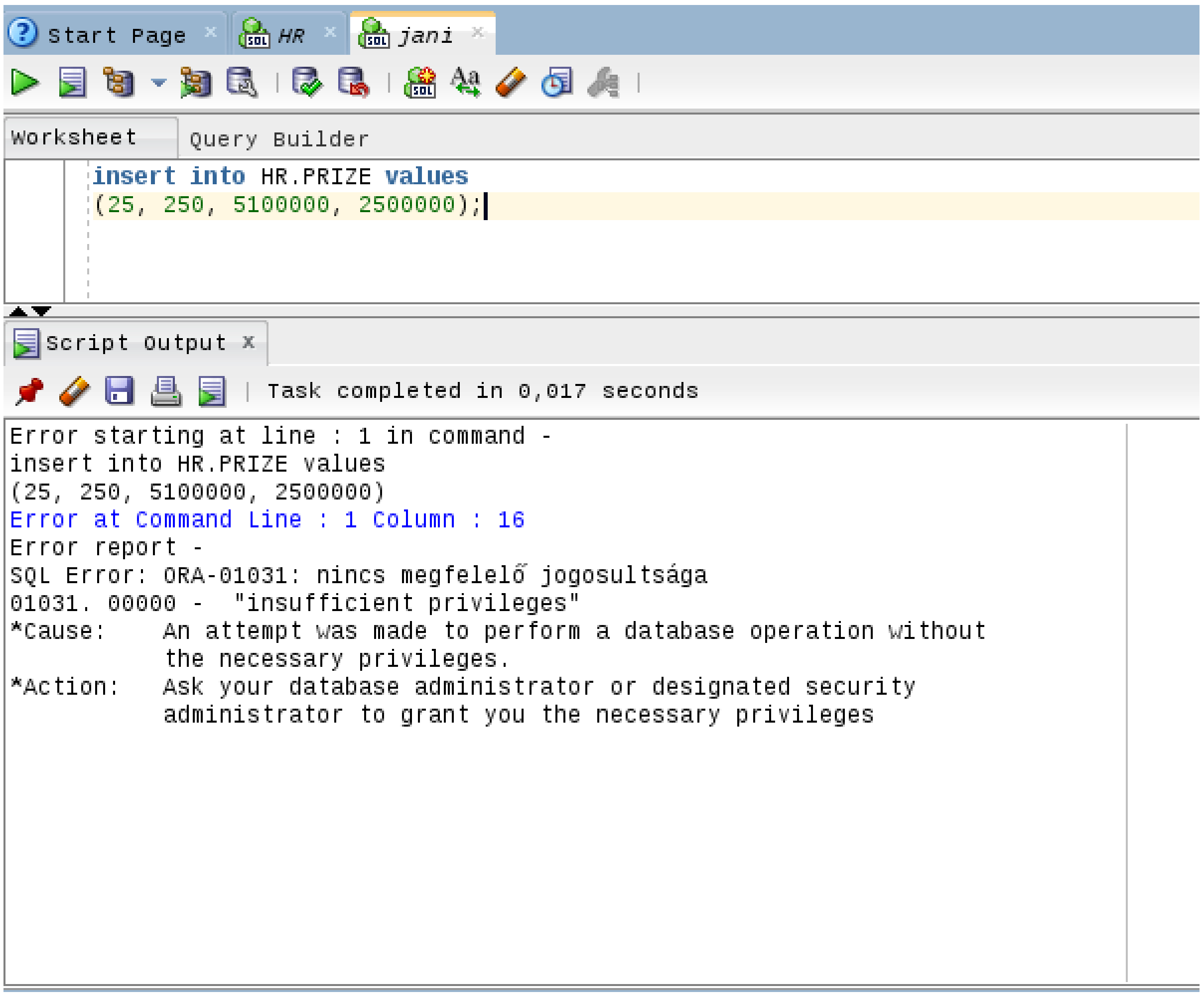


```
insert into HR.singles_champions values
('CA_WE','Caroline Weberer', 'Germany', '2-2', 'F');
```

Task completed in 0,004 seconds

```
1 row inserted.
```

- Fail to insert a new row into the *Prize* table.



```
insert into HR.PRIZE values
(25, 250, 5100000, 2500000);
```

```
Task completed in 0,017 seconds
```

```
Error starting at line : 1 in command -
insert into HR.PRIZE values
(25, 250, 5100000, 2500000)
Error at Command Line : 1 Column : 16
Error report -
SQL Error: ORA-01031: nincs megfelelő jogosultsága
01031. 00000 -  "insufficient privileges"
*Cause:    An attempt was made to perform a database operation without
           the necessary privileges.
*Action:   Ask your database administrator or designated security
           administrator to grant you the necessary privileges
```