

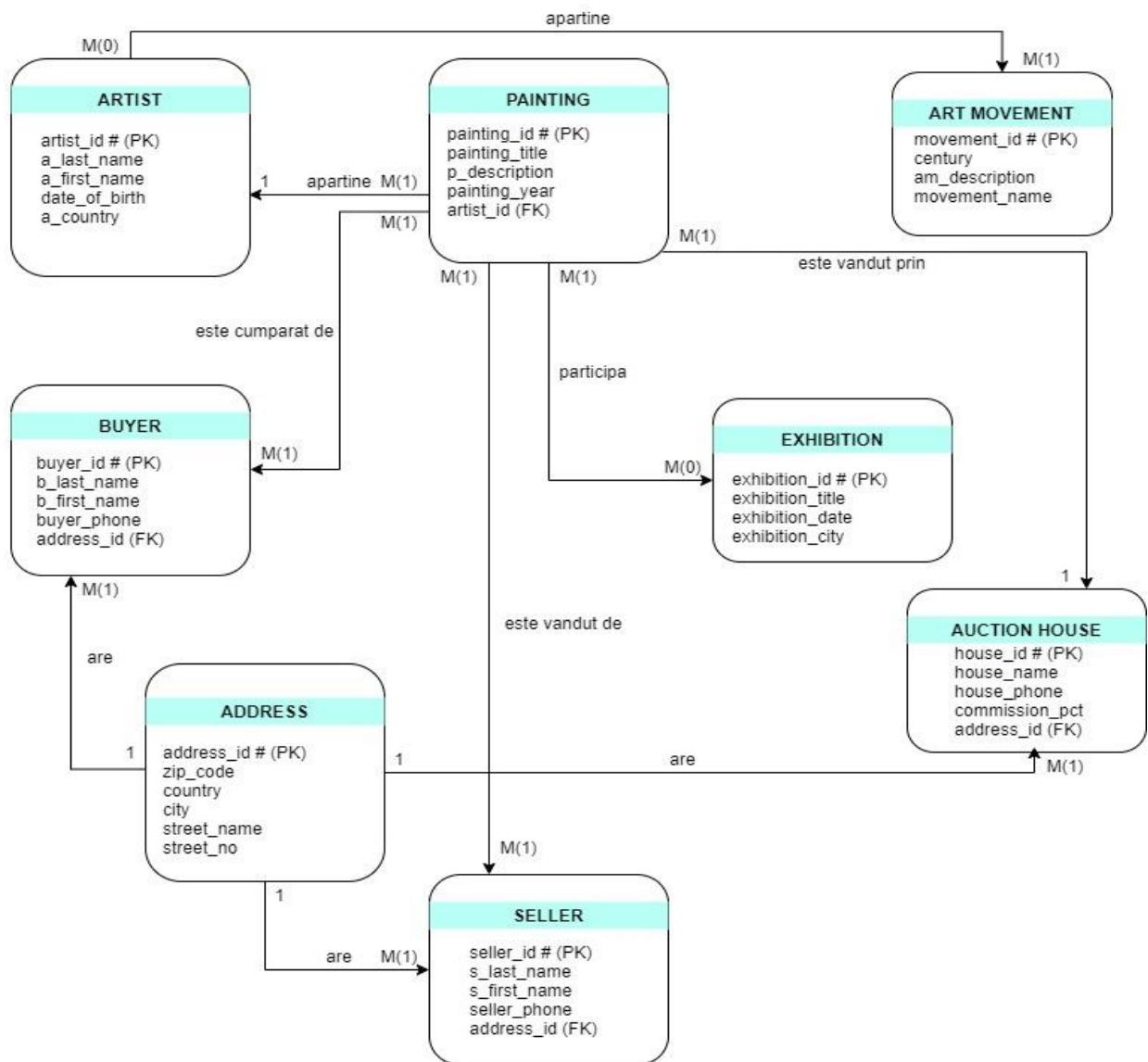
Proiect SGBD

Art Gallery

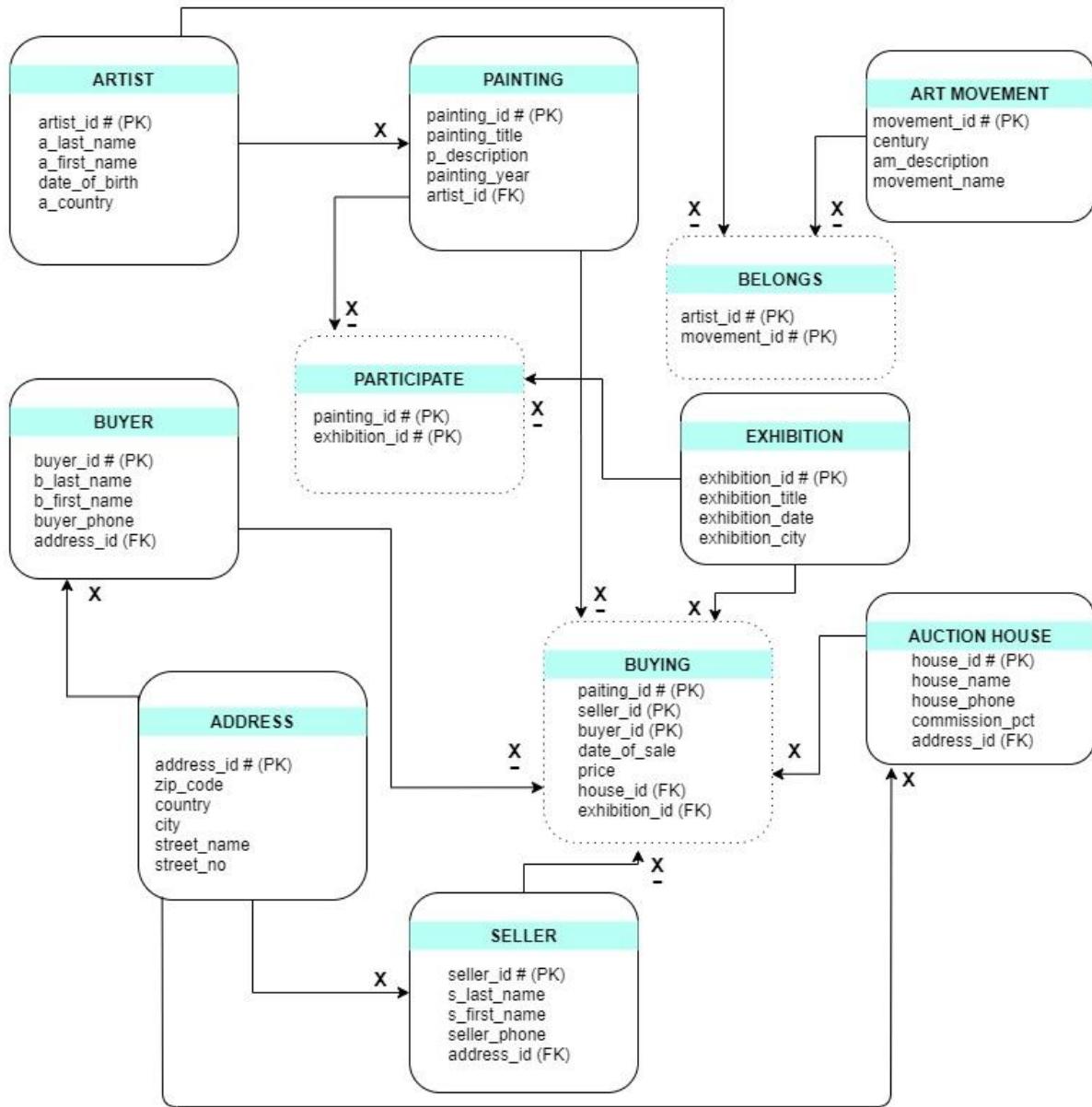
1. Această bază de date a fost gândită cu scopul de a ajuta pasionații de artă plastică să cunoască detalii despre cele mai cunoscute tablouri. Această piață este una destul de volatilă și de interes, aşa că am găsit că ar fi util să existe o bază de date care să stocheze informații despre cele mai impresionante opere de artă.

Pe de altă parte, poate fi posibilă utilizarea acestei baze de date în cadrul unei case de vânzări, pentru a stoca informații despre tablouri, cumpărători, vânzători, achiziții, artiști, curente artistice și expoziții. Stocarea acestor informații ar ajuta casa de vanzari să tina o evidență într-un mod eficient. În scenariul acesta, am considerat ca un artist înregistrat ar fi putut activa în mai multe curente artistice. De asemenea, la o expoziție organizată sunt expuse mai multe tablouri, iar un tablou poate fi expus la mai multe expoziții. Cu toate că aceste bunuri sunt uneori pastrate pe viata, am considerat că în ceea ce privește o achiziție, aceasta este determinată de tabloul care este vândut, de comparator și de vânzator, iar un tablou este vândut în cadrul unei expoziții, printr-o casă de vânzare. Pentru a include și posibilitatea unor achiziții private, am decis că id-ul expoziției și id-ul casei de vanzari să nu fie întotdeauna necesare. De asemenea, am considerat că ar fi util să avem stocate în baza de date adresele cumpărătorilor și vânzătorilor, cât și a caselor de vânzare.

2. Diagrama Entitate-Relație



3. Diagrama Conceptuală



4. Implementarea în Oracle

a. Crearea tabelelor: am început cu tabele care nu aveau chei externe (cele care erau referite de alte tabele), coborând ulterior în “arbore”. Pentru constrangerile referitoare la cheile externe, am folosit in general on delete set null. Am folosit on delete cascade atunci cand cheia primara era compusa din doua chei externe. Am lăsat crearea tabelelor asociative la final.

```
create table artist
(artist_id number(6),
a_last_name varchar2(30) constraint artist_last_name_nn not null,
a_first_name varchar2(30) constraint artist_first_name_nn not null,
date_of_birth date,
a_country varchar2(25),
favourite_style varchar2(30),
constraint artist_artist_id_pk primary key (artist_id));

create table address
(address_id number(6),
zip_code number(11) constraint address_zip_code_nn not null,
country varchar2(25) constraint address_country_nn not null,
city varchar2(25) constraint address_city_nn not null,
street_name varchar2(25),
street_no number(5),
constraint address_address_id_pk primary key (address_id));

create table art_movement
(movement_id number(5),
century varchar2(20),
am_description varchar2(100),
constraint art_movement_id_pk primary key (movement_id));

create table exhibition
(exhibition_id number(5),
exhibition_title varchar2(30),
exhibition_date date constraint exhibition_exhibition_date_nn not null,
address_id number(6),
```

```
constraint exhibition_address_id_fk foreign key (address_id) references
address(address_id) on delete set null,
constraint exhibition_exhibition_id_pk primary key (exhibition_id));

create table auction_house
(house_id number(5),
house_name varchar2(20) constraint a_ahouse_name_nn not null,
house_phone varchar2(10),
address_id number(6),
constraint ah_address_id_fk foreign key (address_id) references
address(address_id) on delete set null,
constraint ah_ahouse_id_pk primary key (house_id));

create table buyer
(buyer_id number(6),
b_last_name varchar2(30) constraint b_last_name_nn not null,
b_first_name varchar2(30) constraint b_first_name_nn not null,
buyer_phone varchar2(10),
address_id number(6),
constraint b_address_id_fk foreign key (address_id) references
address(address_id) on delete set null,
constraint b_buyer_id_pk primary key (buyer_id));

create table seller
(seller_id number(6),
s_last_name varchar2(30) constraint s_last_name_nn not null,
s_first_name varchar2(30) constraint s_first_name_nn not null,
seller_phone varchar2(10),
address_id number(6),
constraint s_seller_id_fk foreign key (address_id) references
address(address_id) on delete set null,
constraint s_seller_id_pk primary key (seller_id));

create table painting
(painting_id number(6),
painting_title varchar2(40),
p_description varchar(100),
status varchar2(12) constraint p_status_nn not null,
```

```
painting_year varchar2(5),  
artist_id number(6),  
constraint s_artist_id_fk foreign key (artist_id) references  
artist(artist_id) on delete set null,  
constraint p_painting_id_pk primary key (painting_id);  
  
create table belongs  
(artist_id number(6),  
movement_id number(5),  
constraint belongs_pk primary key (artist_id, movement_id),  
constraint belongs_artist_id_fk foreign key (artist_id) references  
artist(artist_id) on delete cascade,  
constraint belongs_movement_id_fk foreign key (movement_id) references  
art_movement(movement_id) on delete cascade);  
  
create table participate  
(painting_id number(6),  
exhibition_id number(5),  
constraint participate_pk primary key (painting_id, exhibition_id),  
constraint participate_painting_id_fk foreign key (painting_id) references  
painting(painting_id) on delete cascade,  
constraint participate_exhibition_id_fk foreign key (exhibition_id)  
references exhibition(exhibition_id) on delete cascade);  
  
create table buying  
(painting_id number(6),  
seller_id number(6),  
buyer_id number(6),  
date_of_sale date constraint buying_date_of_sale_nn not null,  
house_id number(5),  
exhibition_id number(5),  
constraint buying_pk primary key (painting_id, seller_id, buyer_id),  
constraint buying_painting_id_fk foreign key (painting_id) references  
painting(painting_id) on delete cascade,  
constraint buying_seller_id_fk foreign key (seller_id) references  
seller(seller_id) on delete cascade,  
constraint buying_buyer_id_fk foreign key (buyer_id) references  
buyer(buyer_id) on delete cascade,  
constraint buying_house_id_fk foreign key (house_id) references  
auction_house(house_id) on delete set null,
```

```
constraint buying_exhibition_id_fk foreign key (exhibition_id) references
exhibition(exhibition_id) on delete set null;
```

b. Alterarea tabelelor: pe parcurs am remarcat îmbunătățiri ce puteau fi aduse sau greșeli. Forma finală este cea prezentată în diagrama conceptuală sau ERD, sau în modelele de mai jos.

```
alter table buying
add price number(11,2);
```

```
alter table auction_house
add commission_pct number(4,2);
```

```
alter table artist
rename column favourite_style to artist_movement;
```

```
alter table art_movement
add movement_name varchar2(30);
```

```
alter table art_movement
modify am_description varchar2(300);
```

```
alter table exhibition
drop column address_id;
```

```
alter table exhibition
add place varchar2(40);
```

```
alter table auction_house
modify house_phone varchar2(12);
```

```
alter table buyer
modify buyer_phone varchar2(12);
```

```
alter table seller
modify seller_phone varchar2(12);
```

```
alter table painting
```

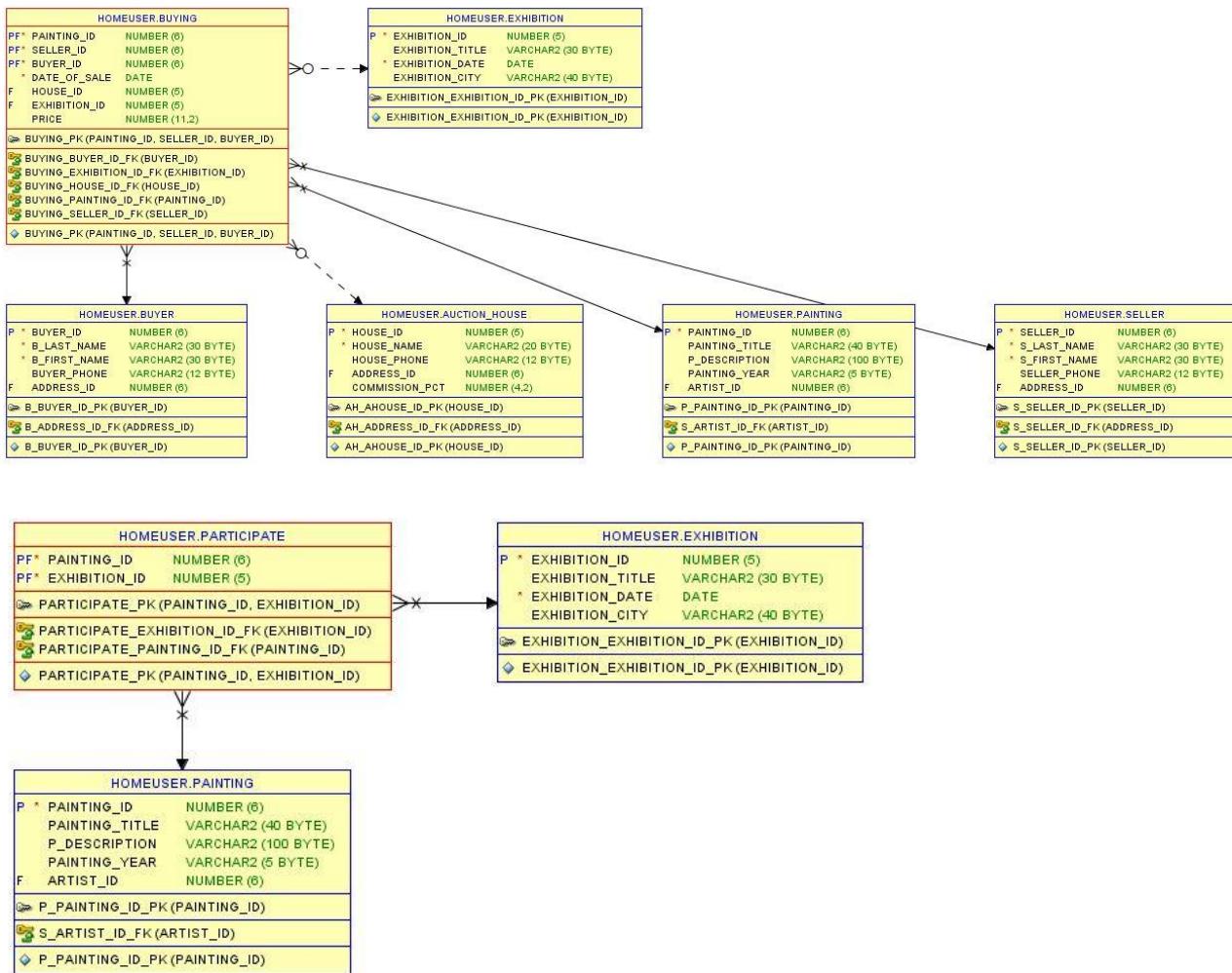
```
drop column status;
```

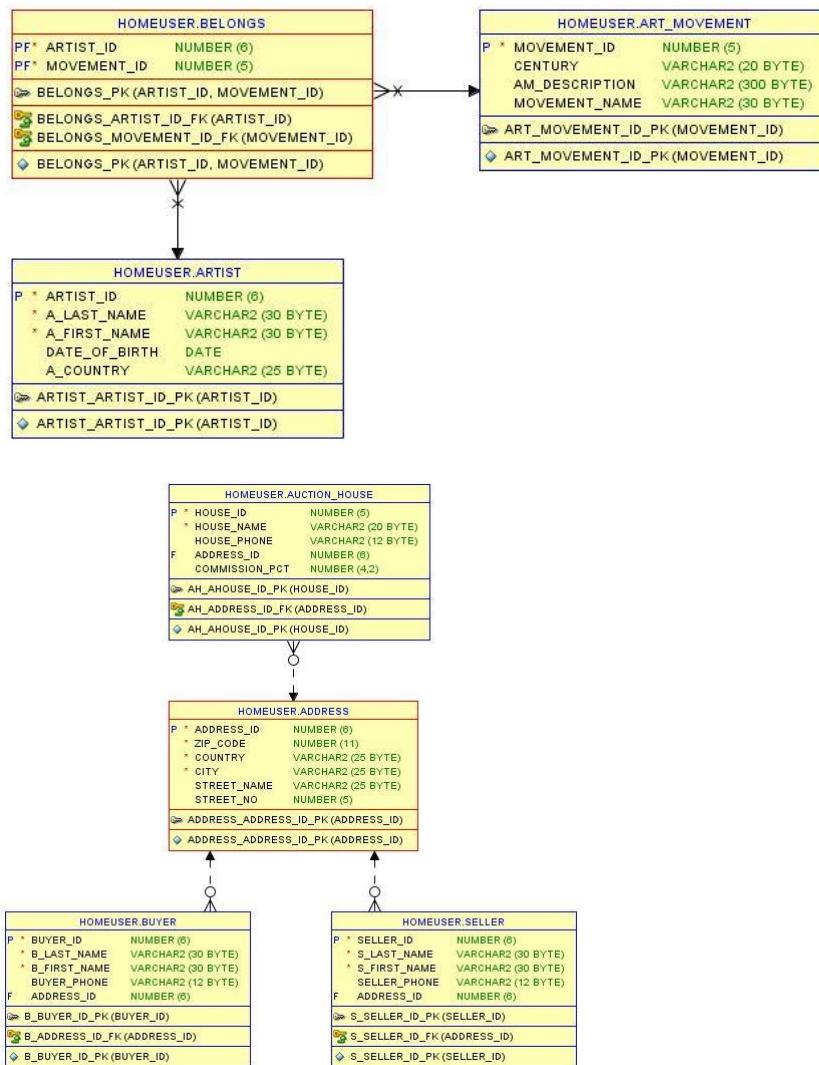
```
alter table artist
```

```
drop column artist_movement;
```

```
alter table exhibition
```

```
rename column city to exhibition_city;
```





5. Popularea tabelelor: am folosit aceeași strategie, inserând întâi valori în tabelele independente.

```

--insert pt artist

insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(1, 'da Vinci', 'Leonardo', to_date('14-04-1452', 'dd-mm-yyyy'), 'Italy');

insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(2, 'Cezanne', 'Paul', to_date('19-01-1839', 'dd-mm-yyyy'), 'France');

insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(3, 'Gauguin', 'Eugene Henri Paul', to_date('07-06-1848', 'dd-mm-yyyy'), 'France');

```

```

insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(4, 'Picasso', 'Pablo Ruiz', to_date('25-10-1881', 'dd-mm-yyyy'), 'Spain' );
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(5, 'de Kooning', 'Williem', to_date('24-04-1904', 'dd-mm-yyyy'), 'Netherlands');
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(6, 'Klimt', 'Gustav', to_date('14-07-1862', 'dd-mm-yyyy'), 'Austria');
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(7, 'Pollock', 'Jackson', to_date('28-01-1912', 'dd-mm-yyyy'), 'USA');
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(8, 'van Gogh', 'Vincent', to_date('30-01-1853', 'dd-mm-yyyy'), 'Netherlands');
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(9, 'Eakins', 'Thomas', to_date('25-07-1844', 'dd-mm-yyyy'), 'USA');
insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country)
values
(10, 'Bacon', 'Francis', to_date('28-10-1909', 'dd-mm-yyyy'), 'Ireland');

```

The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, a script is being run with line numbers 136 through 146. The final command is a SELECT statement:

```

136
137 insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country) values
138 (8, 'van Gogh', 'Vincent', to_date('30-01-1853', 'dd-mm-yyyy'), 'Netherlands');
139
140 insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country) values
141 (9, 'Eakins', 'Thomas', to_date('25-07-1844', 'dd-mm-yyyy'), 'USA' );
142
143 insert into artist (artist_id, a_last_name, a_first_name, date_of_birth, a_country) values
144 (10, 'Bacon', 'Francis', to_date('28-10-1909', 'dd-mm-yyyy'), 'Ireland' );
145
146 select * from artist;
147

```

In the Query Result tab, the output is displayed as a table:

ARTIST_ID	A_LAST_NAME	A_FIRST_NAME	DATE_OF_BIRTH	A_COUNTRY
1	da Vinci	Leonardo	14-04-1452	Italy
2	Cézanne	Paul	19-01-1839	France
3	Gauguin	Eugene Henri Paul	07-06-1848	France
4	Picasso	Pablo Ruiz	25-10-1881	Spain
5	de Kooning	Williem	24-04-1904	Netherlands
6	Klimt	Gustav	14-07-1862	Austria
7	Pollock	Jackson	28-01-1912	USA
8	van Gogh	Vincent	30-01-1853	Netherlands
9	Eakins	Thomas	25-07-1844	USA
10	Bacon	Francis	28-10-1909	Ireland

--insert pt address

```

insert into address(address_id, zip_code, country, city, street_name, street_no)
values

```

```
(1, 00233, 'Italy', 'Bologna', 'Via Barberia', 29);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(2, 29910, 'Italy', 'Florence', 'Via Doccia', 70);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(3, 3233, 'Netherlands', 'Amsterdam', 'Waaigat', 3);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(4, 12344, 'USA', 'New York', 'King St', null);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(5, 3299, 'USA', 'Detroit', 'Franklin St', 143);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(6, 69941, 'Spain', 'Murcia', 'Calle Mar del Caribe', 204);
insert into address(address_id, zip_code, country, city) values

(7, 7203, 'Russia', 'Perm');
insert into address(address_id, zip_code, country, city, street_name) values

(8, 1223, 'Qatar', 'Doha', 'Massafi St');
insert into address(address_id, zip_code, country, city) values

(9, 2345, 'Saudi Arabia', 'Medina');

insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(10, 44509, 'USA', 'Tampa', 'W Spruce St', 503);
insert into address(address_id, zip_code, country, city, street_name) values

(11, 45641, 'Japan', 'Osaka', 'Nipponbashi');
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(12, 32555, 'Mexico', 'Guadalajara', 'Lucas Alaman', 120);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(13, 999221, 'USA', 'Los Angeles', 'Packard St', 223);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(14, 90032, 'USA', 'New York', 'Avenue St', null);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(15, 2113, 'Switzerland', 'Berna', 'Altenbergrain', null);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values

(16, 999234, 'USA', 'Los Angeles', 'New St', 104);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values
```

```
(17, 495300, 'USA', 'Florida', 'Sun St', 11);
insert into address(address_id, zip_code, country, city, street_name, street_no)
values
(18, 22113, 'USA', 'California', 'Palm St', 407);

198 insert into address(address_id, zip_code, country, city, street_name, street_no) values
199 (17, 495300, 'USA', 'Florida', 'Sun St', 11);
200
201 insert into address(address_id, zip_code, country, city, street_name, street_no) values
202 (18, 22113, 'USA', 'California', 'Palm St', 407);
203
204 select * from address;
205
```

	ADDRESS_ID	ZIP_CODE	COUNTRY	CITY	SSTREET_NAME	SSTREET_NO
1	1	233	Italy	Bologna	Via Barberia	29
2	2	29910	Italy	Florence	Via Doccia	70
3	3	3233	Netherlands	Amsterdam	Waaiqat	3
4	4	12344	USA	New York	King St	(null)
5	5	3299	USA	Detroit	Franklin St	143
6	6	69941	Spain	Murcia	Calle Mar del Caribe	204
7	7	7203	Russia	Perm	(null)	(null)
8	8	1223	Qatar	Doha	Massafi St	(null)
9	9	2345	Saudi Arabia	Medina	(null)	(null)
10	10	44509	USA	Tampa	W Spruce St	503
11	11	45641	Japan	Osaka	Nipponbashi	(null)
12	12	32555	Mexico	Guadalajara	Lucas Alaman	120
13	13	999221	USA	Los Angeles	Packard St	223
14	14	90032	USA	New York	Avenue St	(null)
15	15	2113	Switzerland	Berna	Altenbergrain	(null)
16	16	999234	USA	Los Angeles	New St	104
17	17	4953002	USA	Florida	Sun St	11
18	18	22113	USA	CCalifornia	Palm St	407

```
--insert pt art_movement

insert into art_movement(movement_id, century, am_description, movement_name)
values
(1, '15', 'The visual arts of the High Renaissance were marked by a renewed emphasis upon the classical tradition, the expansion of networks of patronage, and a gradual attenuation of figural forms into the style later termed Mannerism.', 'High Renaissance');

insert into art_movement(movement_id, century, am_description, movement_name)
values
(2, '19', 'Is characterized by relatively small, thin, yet visible brush strokes, open composition, emphasis on accurate depiction of light in its changing qualities, ordinary subject matter, inclusion of movement as a crucial element of human perception and experience, and unusual visual angles.', 'Impressionism');

insert into art_movement(movement_id, century, am_description, movement_name)
values
(3, '18', 'The utopian end toward which primitivists aspire usually lies in a notional "state of nature" in which their ancestors existed (chronological primitivism), or in the supposed natural condition of the peoples that live beyond civilization.', 'Primitivism');

insert into art_movement(movement_id, century, am_description, movement_name)
values
(4, '20', 'In Cubist artwork, objects are analyzed, broken up and reassembled in an abstracted form-instead of depicting objects from a single viewpoint, the artist depicts the subject from a multitude of viewpoints to represent the subject in a greater context.', 'Cubism');

insert into art_movement(movement_id, century, am_description, movement_name)
values
```

```
(5, '20', 'The movement name is derived from the combination of the emotional intensity and self-denial of the German Expressionists with the anti-figurative aesthetic of the European abstract schools such as Futurism, the Bauhaus, and Synthetic Cubism.', 'Abstract Expressionism');

insert into art_movement(movement_id, century, am_description, movement_name)
values

(6, '19', 'Stood for idealism, sentimentalism or exoticism, alongside a noted interest in spirituality and esotericism.', 'Symbolism');

insert into art_movement(movement_id, century, am_description, movement_name)
values

(7, '19', 'Wanted to break down the traditional distinction between fine arts and applied arts.', 'Art Nouveau');

insert into art_movement(movement_id, century, am_description, movement_name)
values

(8, '19', 'Created a "total art", that unified painting, architecture, and the decorative arts.', 'Vienna Secession');

insert into art_movement(movement_id, century, am_description, movement_name)
values

(9, '19', 'Post-Impressionism emerged as a reaction against Impressionists concern for the naturalistic depiction of light and colour.', 'Post-Impressionism');

insert into art_movement(movement_id, century, am_description, movement_name)
values

(10, '20', 'Known for juxtaposition of uncommon imagery.', 'Surrealism');
```

The screenshot shows a MySQL query editor with two tabs: 'Query' and 'Result'. The 'Query' tab contains the SQL code for creating the 'art_movement' table and inserting 10 rows of data. The 'Result' tab displays the 10 rows of data in a table format.

MOVEMENT_ID	MOVEMENT_NAME	CENTURY	AM_DESCRIPTION
1	High Renaissance	15	The visual arts of the High Renaissance were marked by a renewed
2	Impressionism	19	Is characterized by relatively small, thin, yet visible brush str
3	Primitivism	18	The utopian end toward which primitivists aspire usually lies in
4	Cubism	20	In Cubist artwork, objects are analyzed, broken up and reassemble
5	Abstract Expressionism	20	The movement name is derived from the combination of the emotiona
6	Symbolism	19	Stood for idealism, sentimentalism or exoticism, alongside a note
7	Art Nouveau	19	Wanted to break down the traditional distinction between fine art
8	Vienna Secession	19	Created a "total art", that unified painting, architecture, and t
9	Post-Impressionism	19	Post-Impressionism emerged as a reaction against Impressionists c
10	Surrealism	20	Known for juxtaposition of uncommon imagery.

```
--insert pt exhibition

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values

(1, '2001 Best Paintings', to_date('28-01-2001', 'dd-mm-yyyy'), 'New York');

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values

(2, 'In The Making', to_date('09-11-2019', 'dd-mm-yyyy'), 'Tokio');
```

```

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values
(3, 'Shape Lab', to_date('11-03-2020', 'dd-mm-yyyy'), 'California');

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values
(4, 'Building Collections', to_date('29-05-2018', 'dd-mm-yyyy'), 'New York');

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values
(5, 'American Modern', to_date('30-09-2017', 'dd-mm-yyyy'), 'Los Angeles');

insert into exhibition(exhibition_id, exhibition_title, exhibition_date, place)
values
(6, 'Sunflower', to_date('04-01-2020', 'dd-mm-yyyy'), 'California');

```

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'Worksheet' and contains a SQL query with numbered lines from 268 to 277. Lines 268 through 275 show the insertion of six exhibition records. Line 276 is a 'select * from exhibition;' statement highlighted in yellow. The bottom window is titled 'Query Result' and displays the results of the query as a table. The table has columns: EXHIBITION_ID, EXHIBITION_TITLE, EXHIBITION_DATE, and EXHIBITION_CITY. The data is as follows:

EXHIBITION_ID	EXHIBITION_TITLE	EXHIBITION_DATE	EXHIBITION_CITY
1	12001 Best Paintings	28-01-2001	New York
2	2In The Making	09-11-2019	Tokio
3	3Shape Lab	11-03-2020	California
4	4Building Collections	29-05-2018	New York
5	5American Modern	30-09-2017	Los Angeles
6	6Sunflower	04-01-2020	California

--insert pt auction house

```

insert into auction_house(house_id, house_name, house_phone, address_id,
commission_pct) values
(1, 'Christie','917-277-1151', 4, 5);

insert into auction_house(house_id, house_name, house_phone, address_id,
commission_pct) values
(2, 'Larry Gagosian','886-300-1100', 3, 0.5);

insert into auction_house(house_id, house_name, house_phone, address_id,
commission_pct) values
(3, 'Sotheby','223-299-1212', 10, 4.8);

insert into auction_house(house_id, house_name, house_phone, address_id,
commission_pct) values
(4, 'Steven Mazoh', '917-277-1151', 13, 3.9);

```

```

Worksheet | Query Builder
286 insert into auction_house(house_id, house_name, house_phone, address_id, commission_pct) values
287 (2, 'Larry Gagosian','886-300-1100', 3, 5);
288
289 insert into auction_house(house_id, house_name, house_phone, address_id, commission_pct) values
290 (3, 'Sotheby','223-299-1212', 10, 4.8);
291
292 insert into auction_house(house_id, house_name, house_phone, address_id, commission_pct) values
293 (4, 'Steven Mazoh', '917-277-1151', 13, 3.9);
294
295 select * from auction_house;

```

Query Result x | All Rows Fetched: 4 in 0.012 seconds

HOUSE_ID	HOUSE_NAME	HOUSE_PHONE	ADDRESS_ID	COMMISSION_PCT
1	Christie	917-277-1151	4	5
2	Larry Gagosian	886-300-1100	5	0,5
3	Sotheby	223-299-1212	10	4,8
4	Steven Mazoh	917-277-1151	13	3,9

--insert pt buyer

```

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(1, 'Al Saud', 'Badr bin Abdullah', '223-008-0092', 8);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(2, 'Griffin', 'Kenneth', '890-808-123', 14);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(3, 'Rybolovlev', 'Dmitry', '700-122-2003', 7);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(4, 'Liu', 'Yiqian', '301-177-8880', 11);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(5, 'Martinez', 'David', '455-420-2020', 12);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(6, 'Staechelin', 'Rudolf', '233-009-1909', 5);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(7, 'Al Thani', 'Hamad', null, 9);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(8, 'James', 'Steven', null, 16);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(9, 'Wynn', 'Steve', null, 17);

insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id)
values
(10, 'Whitney', 'Betsey', null, 18);

```

The screenshot shows a MySQL Workbench session. The 'Worksheet' tab contains the following SQL code:

```

323 insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id) val
324 (8, 'James', 'Steven', null, 16);
325
326 insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id) val
327 (9, 'Wynn', 'Steve', null, 17);
328
329 insert into buyer(buyer_id, b_last_name, b_first_name, buyer_phone, address_id) val
330 (10, 'Whitney', 'Betsey', null, 18);
331
332 select * from buyer;

```

The 'Query Result' tab displays the output of the last query, showing 10 rows of data:

BUYER_ID	B_LAST_NAME	B_FIRST_NAME	BUYER_PHONE	ADDRESS_ID
1	Staechelin	Rudolf	233-009-1909	5
2	Al Thani	Hamad	(null)	9
3	James	Steven	(null)	16
4	Wynn	Steve	(null)	17
5	Whitney	Betsey	(null)	18
6	Al Saud	Badr bin Abdullah	223-008-0092	8
7	Griffin	Kenneth	890-808-123	14
8	Rybolovlev	Dmitry	700-122-2003	7
9	Liu	Yiqian	301-177-8880	11
10	Martinez	David	455-420-2020	12

--insert pt seller

```

insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id)
values
(1, 'Rybolovlev', 'Dmitry', '700-122-2003', 7);

insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id)
values
(2, 'Geffen', 'David', '566-444-2443', 1);

insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id)
values
(3, 'Embirkicos', 'George', '333-231-2213', 6);

insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id)
values
(4, 'Staechelin', 'Rudolf', '921-219-7004', 15);

insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id)
values
(5, 'Altmann', 'Maria', '521-400-6399', 2);

```

The screenshot shows a MySQL Workbench interface. The top part is a 'Query Builder' window containing the following SQL code:

```

347
348 insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id) values
349 (4, 'Staechelin', 'Rudolf', '921-219-7004', 15);
350
351 insert into seller(seller_id, s_last_name, s_first_name, seller_phone, address_id) values
352 (5, 'Altmann', 'Maria', '521-400-6399', 2);
353
354 select * from seller;
355

```

The bottom part is a 'Query Result' window showing the output of the query:

SELLER_ID	S_LAST_NAME	S_FIRST_NAME	SELLER_PHONE	ADDRESS_ID
1	Rybolovlev	Dmitry	700-122-2003	7
2	Geffen	David	566-444-2443	1
3	Embiricos	George	333-231-2213	6
4	Staechelin	Rudolf	921-219-7004	15
5	Altmann	Maria	521-400-6399	2

All Rows Fetched: 5 in 0.004 seconds

```
--insert pt painting

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(1, 'Salvador Mundi', 'Jesus in Renaissance dress, making the sign of the cross.', 'sold', 1500, 1);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(2, 'The Card Players', 'The men look down at their cards rather than at each other.', 'sold', 1892, 3);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(3, 'When Will You Marry?', 'The front and middle ground are built up in areas of green, yellow and blue.', 'sold', 1892, 3);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(4, 'Women of Algiers', 'Image of languid, voluptuous women known as odalisques.', 'sold', 1955, 4);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(5, 'Interchange', 'It was one of de Kooning first abstract landscapes.', 'sold', 1955, 5);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(6, 'Boy with a Pipe', 'A Parisian adolescent boy who holds a pipe in his left hand.', 'available', 1905, 4);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(7, 'Woman III', ' Done between 1951 and 1953 in which the central theme was a woman.', 'available', 1953, 5);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(8, 'Le Reve', 'Is portraying his 22-year-old mistress Marie-Thérèse Walter.', 'available', 1932, 4);
```

```
insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(9, 'Otahi', 'Also known as Alone.', 'available', 1893, 3);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(10, 'Nude, Green Leaves and Bust', 'Featuring his mistress Marie-Thérèse Walter.', 'available', 1932, 4);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(11, 'Portrait of Adele Bloch-Bauer I', 'Also called The Lady in Gold or The Woman in Gold.', 'sold', 1907, 6);

insert into painting(painting_id, painting_title, p_description, status,
painting_year, artist_id) values
(12, 'No. 5, 1948', 'The painting was created on fibreboard.', 'sold', 1948, 7);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(13, 'Portrait of Adele Bloch-Bauer II', 'Adele Bloch-Bauer was the only person whose portrait was painted twice by Klimt.', 1912, 6);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(14, 'Dora Maar au Chat', 'Dora Maar au Chat is one of the largest portraits of the subject by Picasso.', 1941, 4);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(15, 'Young Girl with a Flower Basket', 'The subject of the painting is a young girl who was working as a flower seller.', 1905, 4);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(16, 'Les Noces de Pierrette', 'The marriage of Pierrette.', 1905, 4);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(17, 'Yo, Picasso', 'Portret.', 1901, 4);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(18, 'Water Serpents II', 'It is the follow-up painting to the earlier painting Water Serpents I.', 1904, 6);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(19, 'Rideau, Cruchon et Compotier', 'It is considered the most expensive still life ever sold at an auction..', 1893, 2);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(20, 'Police Gazette', 'A landscape painted on canvas using abstract elements, and colors such as yellow, green and red.', 1955, 5);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(21, 'Portrait of Dr. Gachet', 'Shows Gachet sitting at a table and leaning his head on his right arm.', 1890, 8);
```

```

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(22, 'Portrait of the Postman Joseph Roulin', 'Also known as The Father.', 1888,
8);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(23, 'Irises', 'The painting is full of softness and lightness.', 1889, 8);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(24, 'Starry Night Over the Rhône', 'The sky is aquamarine, the water is royal
blue.', 1888, 8);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(25, 'Self-portrait without beard', 'This may have been his last self-portrait.',
1889, 8);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(26, 'Gioconda', 'Also known as Mona Lisa.', 1503, 1);

insert into painting(painting_id, painting_title, p_description, painting_year,
artist_id) values
(27, 'Triptych, 1976', 'It is the second most expensive Bacon ever sold.', 1976,
10);

```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a 'Worksheet' tab and a 'Query Builder' tab. Below these tabs, the code for inserting data into the 'painting' table is visible, spanning lines 437 to 441. The 'select * from painting;' command is highlighted in blue. In the bottom-right corner, there's a 'Query Result' tab. It displays a table with 27 rows, each corresponding to a painting from the list above. The columns are labeled: PAINTING_ID, PAINTING_TITLE, PAINTING_YEAR, ARTIST_ID, and P_DESCRIPTION. The table content is as follows:

PAINTING_ID	PAINTING_TITLE	PAINTING_YEAR	ARTIST_ID	P_DESCRIPTION
1	Salvador Mundi	1500	1	Jesus in Renaissance dress, making th-
2	The Card Players	1892	3	The men look down at their cards rath-
3	When Will You Marry?	1892	3	The front and middle ground are built
4	Women of Algiers	1955	4	Image of languid, voluptuous women kr
5	Interchange	1955	5	It was one of de Kooning first abstra
6	Boy with a Pipe	1905	4	A Parisian adolescent boy who holds a
7	Woman III	1953	5	Done between 1951 and 1953 in which
8	Le Reve	1932	4	Is portraying his 22-year-old mistres
9	Otahí	1893	3	Also known as Alone.
10	Nude, Green Leaves and Bust	1932	4	Featuring his mistress Marie-Thérèse
11	Portrait of Adele Bloch-Bauer I	1907	6	Also called The Lady in Gold or The W
12	No. 5, 1948	1948	7	The painting was created on fibreboar
13	Portrait of Adele Bloch-Bauer II	1912	6	Adele Bloch-Bauer was the only person
14	Dora Maar au Chat	1941	4	Dora Maar au Chat is one of the large
15	Young Girl with a Flower Basket	1905	4	The subject of the painting is a your
16	Les Noces de Pierrette	1905	4	The marriage of Pierrette.
17	Yo, Picasso	1901	4	Portret.
18	Water Serpents II	1904	6	It is the follow-up painting to the e
19	Rideau, Cruchon et Compotier	1893	2	It is considered the most expensive s
20	Police Gazette	1955	5	A landscape painted on canvas using a
21	Portrait of Dr. Gachet	1890	8	Shows Gachet sitting at a table and i
22	Portrait of the Postman Joseph Roulin	1888	8	Also known as The Father.
23	Irises	1889	8	The painting is full of softness and
24	Starry Night Over the Rhône	1888	8	The sky is aquamarine, the water is r
25	Self-portrait without beard	1889	8	This may have been his last self-port
26	Gioconda	1503	1	Also known as Mona Lisa.
27	Triptych, 1976	1976	10	It is the second most expensive Bacon

--insert pt participate

```
insert into participate(painting_id, exhibition_id) values (1, 2);
```

```
insert into participate(painting_id, exhibition_id) values (2, 4);
insert into participate(painting_id, exhibition_id) values (3, 5);
insert into participate(painting_id, exhibition_id) values (4, 3);
insert into participate(painting_id, exhibition_id) values (5, 2);
insert into participate(painting_id, exhibition_id) values (6, 5);
insert into participate(painting_id, exhibition_id) values (7, 3);
insert into participate(painting_id, exhibition_id) values (8, 3);
insert into participate(painting_id, exhibition_id) values (9, 3);
insert into participate(painting_id, exhibition_id) values (10, 4);
insert into participate(painting_id, exhibition_id) values (10, 5);
insert into participate(painting_id, exhibition_id) values (8, 2);
insert into participate(painting_id, exhibition_id) values (8, 4);
insert into participate(painting_id, exhibition_id) values (27, 1);
insert into participate(painting_id, exhibition_id) values (27, 1);
insert into participate(painting_id, exhibition_id) values (26, 1);
insert into participate(painting_id, exhibition_id) values (11, 1);
insert into participate(painting_id, exhibition_id) values (12, 3);
insert into participate(painting_id, exhibition_id) values (12, 1);
insert into participate(painting_id, exhibition_id) values (13, 3);
insert into participate(painting_id, exhibition_id) values (14, 5);
insert into participate(painting_id, exhibition_id) values (15, 2);
insert into participate(painting_id, exhibition_id) values (16, 2);
insert into participate(painting_id, exhibition_id) values (17, 4);
insert into participate(painting_id, exhibition_id) values (18, 4);
insert into participate(painting_id, exhibition_id) values (19, 5);
insert into participate(painting_id, exhibition_id) values (20, 1);
insert into participate(painting_id, exhibition_id) values (21, 1);
insert into participate(painting_id, exhibition_id) values (21, 3);
insert into participate(painting_id, exhibition_id) values (22, 5);
insert into participate(painting_id, exhibition_id) values (23, 4);
insert into participate(painting_id, exhibition_id) values (24, 4);
insert into participate(painting_id, exhibition_id) values (25, 2);
insert into participate(painting_id, exhibition_id) values (1, 5);
insert into participate(painting_id, exhibition_id) values (21, 4);
insert into participate(painting_id, exhibition_id) values (18, 2);
insert into participate(painting_id, exhibition_id) values (13, 1);
insert into participate(painting_id, exhibition_id) values (4, 5);
insert into participate(painting_id, exhibition_id) values (12, 4);
```

```

insert into participate(painting_id, exhibition_id) values (17, 2);
insert into participate(painting_id, exhibition_id) values (19, 2);
insert into participate(painting_id, exhibition_id) values (20, 2);
insert into participate(painting_id, exhibition_id) values (11, 2);
insert into participate(painting_id, exhibition_id) values (1, 1);
insert into participate(painting_id, exhibition_id) values (2, 5);
insert into participate(painting_id, exhibition_id) values (3, 2);
insert into participate(painting_id, exhibition_id) values (5, 4);
insert into participate(painting_id, exhibition_id) values (6, 3);
insert into participate(painting_id, exhibition_id) values (9, 5);
insert into participate(painting_id, exhibition_id) values (13, 2);
insert into participate(painting_id, exhibition_id) values (15, 5);
insert into participate(painting_id, exhibition_id) values (23, 2);
insert into participate(painting_id, exhibition_id) values (27, 5);
insert into participate(painting_id, exhibition_id) values (14, 1);

```

PAINTING_ID	EXHIBITION_ID
23	11
24	12
25	12
26	12
27	13
28	13
29	13
30	14
31	14
32	15
33	15
34	16
35	17
36	17
37	18
38	18
39	19
40	19
41	20
42	20
43	21
44	21
45	21
46	22
47	23
48	23
49	24
50	25
51	26
52	27
53	27

--insert pt buying

```

insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(1, 1, 1, to_date('15-11-2017', 'dd-mm-yyyy'), 1, 1, 469.7);

```

```
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(2, 3, 1, to_date('15-08-2015', 'dd-mm-yyyy'), 3, 5, 284);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(3, 4, 2, to_date('03-08-2014', 'dd-mm-yyyy'), 4, 2, 227);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(4, 4, 2, to_date('03-08-2014', 'dd-mm-yyyy'), 1, 3, 193.5);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(5, 2, 2, to_date('11-05-2015', 'dd-mm-yyyy'), 2, 4, 324);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(6, 2, 3, to_date('02-11-2006', 'dd-mm-yyyy'), 3, 3, 141);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(7, 3, 4, to_date('02-11-2006', 'dd-mm-yyyy'), 3, 3, 177.6);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(8, 3, 1, to_date('26-03-2013', 'dd-mm-yyyy'), 3, 3, 170.1);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(9, 5, 3, to_date('24-08-2013', 'dd-mm-yyyy'), 3, 5, 131.7);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(10, 5, 6, to_date('04-05-2010', 'dd-mm-yyyy'), 3, 4, 124.8);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(11, 4, 7, to_date('11-05-2015', 'dd-mm-yyyy'), 1, 2, 171.2);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(12, 4, 7, to_date('02-11-2006', 'dd-mm-yyyy'), 3, 1, 177.6);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(13, 1, 4, to_date('23-02-2016', 'dd-mm-yyyy'), 2, 2, 159.8);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(14, 2, 5, to_date('23-02-2016', 'dd-mm-yyyy'), 3, 5, 120.8);

insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
(15, 4, 2, to_date('08-05-2018', 'dd-mm-yyyy'), 3, 5, 117.1);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values
```

```

(16, 5, 7, to_date('30-11-2000', 'dd-mm-yyyy'), 4, 2, 101.7);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(17, 2, 6, to_date('30-11-2000', 'dd-mm-yyyy'), 4, 2, 101.7);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(18, 5, 3, to_date('13-04-2013', 'dd-mm-yyyy'), null, null, 201.7);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(19, 1, 7, to_date('10-05-1999', 'dd-mm-yyyy'), null, null, 92.9);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(20, 2, 5, to_date('12-10-2006', 'dd-mm-yyyy'), null, null, 80.5);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(21, 5, 6, to_date('15-05-1990', 'dd-mm-yyyy'), null, null, 161.4);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(22, 1, 4, to_date('15-05-1990', 'dd-mm-yyyy'), null, null, 123.8);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(23, 2, 2, to_date('11-11-1987', 'dd-mm-yyyy'), 4, 2, 121.3);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(24, 5, 6, to_date('11-11-1987', 'dd-mm-yyyy'), null, null, 101.3);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(25, 4, 7, to_date('11-11-1987', 'dd-mm-yyyy'), null, null, 71.5);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(26, 2, 3, to_date('11-11-1987', 'dd-mm-yyyy'), null, null, 860);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(27, 5, 10, to_date('14-05-2018', 'dd-mm-yyyy'), 3, 5, 102.5);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(11, 2, 8, to_date('11-12-2019', 'dd-mm-yyyy'), 2, 1, 190.4);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(14, 5, 10, to_date('03-05-2019', 'dd-mm-yyyy'), 4, 1, 130.9);
insert into buying(painting_id, seller_id, buyer_id, date_of_sale, house_id,
exhibition_id, price) values

(3, 4, 9, to_date('21-01-2019', 'dd-mm-yyyy'), 3, 2, 250.4 );

```

Worksheet | Query Builder

```

717
718 select * from buying;
719
720

```

Query Result | All Rows Fetched: 30 in 0,004 seconds

	PAINTING_ID	PRICE	SELLER_ID	BUYER_ID	DATE_OF_SALE	HOUSE_ID	EXHIBITION_ID
1	1	469,7	1	1	15-11...	1	1
2	2	284	3	1	15-08...	3	5
3	3	227	4	2	03-08...	4	2
4	3	250,4	4	9	21-01...	3	2
5	4	193,5	4	2	03-08...	1	3
6	5	324	2	2	11-05...	2	4
7	6	141	2	2	02-11...	3	3
8	7	177,6	3	4	02-11...	3	3
9	8	170,1	3	1	26-03...	3	3
10	9	131,7	5	3	24-08...	3	5
11	10	124,8	5	6	04-05...	3	4
12	11	190,4	2	8	11-12...	2	1
13	11	171,2	5	3	11-05...	1	2
14	12	177,6	2	5	02-11...	3	1
15	13	159,8	1	4	23-02...	2	2
16	14	120,8	2	5	23-02...	3	5
17	14	130,9	5	10	03-05...	4	1
18	15	117,1	4	2	08-05...	3	5
19	16	101,7	5	7	30-11...	4	2
20	17	98,7	2	6	30-11...	4	2
21	18	201,7	5	3	13-04...	(null)	(null)
22	19	92,9	1	7	10-05...	(null)	(null)
23	20	80,5	2	5	12-10...	(null)	(null)
24	21	161,4	5	6	15-05...	(null)	(null)
25	22	123,8	1	4	15-05...	(null)	(null)
26	23	121,3	2	2	11-11...	4	2
27	24	101,3	5	6	11-11...	(null)	3
28	25	71,5	4	7	11-11...	(null)	(null)
29	26	860	2	3	11-11...	(null)	(null)
30	27	102,5	5	10	14-05...	3	5

```
--insert pt belongs

insert into belongs(artist_id, movement_id) values(1, 1);
insert into belongs(artist_id, movement_id) values(2, 2);
insert into belongs(artist_id, movement_id) values(2, 9);
insert into belongs(artist_id, movement_id) values(3, 9);
insert into belongs(artist_id, movement_id) values(3, 3);
insert into belongs(artist_id, movement_id) values(4, 4);
insert into belongs(artist_id, movement_id) values(4, 10);
insert into belongs(artist_id, movement_id) values(5, 5);
insert into belongs(artist_id, movement_id) values(6, 6);
insert into belongs(artist_id, movement_id) values(6, 7);
insert into belongs(artist_id, movement_id) values(6, 8);
insert into belongs(artist_id, movement_id) values(7, 5);
insert into belongs(artist_id, movement_id) values(8, 9);
insert into belongs(artist_id, movement_id) values(10, 4);
insert into belongs(artist_id, movement_id) values(10, 5);
```

```

Worksheet | Query Builder
773
774 insert into belongs(artist_id, movement_id) values
775 (10, 4);
776
777 insert into belongs(artist_id, movement_id) values
778 (10, 5);
779
780 select * from belongs;
781

```

Query Result | All Rows Fetched: 15 in 0,008 seconds

ARTIST_ID	MOVEMENT_ID
1	1
2	2
3	2
4	3
5	3
6	4
7	4
8	5
9	6
10	6
11	6
12	7
13	8
14	10
15	10

6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul:

Pentru a determina cat de reusita a fost fiecare expozitie, ne dorim sa stim cate tablouri au fost expuse in fiecare expozitie si cate dintre acestea au fost vandute. Acest lucru se va realiza prin determinarea unui procent care reprezinta cat % din tablourile expuse au fost vandute in cadrul fiecarei expozitii. Retineti intr-un nou tabel al expozitiilor, pe langa informatiile anterioare, cate tablouri au fost expuse, cate au fost vandute si care este acel procent.

```
--Vom face un vector de record in care retinem pt fiecare expozitie: titlul,
--nr de tablouri expuse, nr de tablouri vandute si procentul
--in prima faza vom pune titlul expozitiei si tablourile expuse luate din tabelul
--asociativ participate, ulterior luand din buying cate au fost vandute si
--calculand acel procent.
```

```
create table exhibition_ado as
select * from exhibition;
```

```
alter table exhibition_ado
add nr_tablouri number(2);
```

```
alter table exhibition_ado
add nr_tablouri_vandute number(2);

alter table exhibition_ado
add raport_tab number(5,2);

create or replace procedure proc_ex6
is
type tabells is record(titlu exhibition.exhibition_title%type,
                      nrTab number(2),
                      nrTabV number (2),
                      rap number(5,2));
type vector_tabl is varray(20) of tabells;

t vector_tabl := vector_tabl();
nr number(2);
r number(5,2);

begin
  select e.exhibition_title, count(*), 0, 0
  bulk collect into t
  from exhibition e, participate p
  where p.exhibition_id = e.exhibition_id
  group by e.exhibition_title;

  for i in t.first..t.last
  loop
    select count(b.exhibition_id)
    into nr
    from exhibition e, buying b
    where e.exhibition_id = b.exhibition_id
    and e.exhibition_title = t(i).titlu
    group by e.exhibition_title;

    if t(i).nrTab != 0
    then
      r := t(i).nrTabV * 100 / t(i).nrTab;
      t(i).rap := r;
    end if;
```

```

end loop;

for i in 1..t.last
loop
    dbms_output.put_line('Titlu: ' || t(i).titlu);
    dbms_output.put_line('Nr de tablouri ' || ' ' || t(i).nrTab);
    dbms_output.put_line('Nr de tablouri vandute ' || ' ' || t(i).nrTabV);
    dbms_output.put_line('Au fost vandute ' || t(i).rap || '% din tablourile
expuse.');
    dbms_output.new_line();

update exhibition_ado
set nr_tablouri = t(i).nrTab
where exhibition_title = t(i).titlu;

update exhibition_ado
set nr_tablouri_vandute = t(i).nrTabV
where exhibition_title = t(i).titlu;

update exhibition_ado
set raport_tab = t(i).rap
where exhibition_title = t(i).titlu;
end loop;
end proc_ex6;
/
execute proc_ex6;
/

--O mica analiza asupra bazei noastre de date:
--1: 2001: 9 tablouri au fost expuse, 4 vandute
--4: building: 10 tablouri au fost expuse, 2 vandute
--5: american: 12 tablouri au fost expuse, 5 vandute
--3: shape: 8 tablouri au fost expuse, 4 vandute
--2: in the: 14 tablouri au fost expuse, 7 vandute

select * from exhibition_ado;

```

The screenshot shows the Oracle SQL Developer interface. On the left, a code editor displays a PL/SQL procedure named 'proc_ex6'. The procedure performs several updates on the 'exhibition_ado' table based on values from a cursor variable 't(i)'. It then executes a select statement to retrieve all rows from 'exhibition_ado'. On the right, the results of the query are displayed in a tree view. The results table shows five rows of data:

EXHIBITION_ID	EXHIBITION_TITLE	EXHIBITION_DATE	EXHIBITION_CITY	NP_TABLOURI	NP_TABLOURI_VANDUTE	RAPORT_TAB
1	1 2001 Best Paintings	28-01-2001	New York	9	4	44,44
2	2 In The Making	09-11-2019	Tokio	14	7	50
3	3 Shape Lab	11-03-2020	California	8	5	62,5
4	4 Building Collections	29-05-2018	New York	10	2	20
5	5 American Modern	30-09-2017	Los Angeles	12	5	41,67

7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul:

Se dorește o statistică asupra artistilor. Pentru fiecare artist, afișați în cele curente artistice a creat și care sunt cele mai scumpe 3 tablouri ale sale.

Tablourile trebuie afișate cu număr de ordine, în ordinea descrescătoare a prețurilor lor.

In cazul în care un artist nu figurează cu cel puțin 3 tablouri, sau nu are înregistrat nici un tablou, menționați acest lucru.

Pentru artiștii care figurează cu cel puțin 3 tablouri afișați și cat valoarea în total a acestor 3 tablouri.

```
--vom folosi un tabel indexat de recorduri în care vom reține titlul și pretul unui tablou
```

```
--într-un alt tablou indexat, avem recorduri pentru fiecare current
```

```
--cursorul îl vom folosi pentru a parcurge numele și prenumele artiștilor
```

```
--pentru a lua top 3 cele mai scumpe tablouri, facem o subcerere în from
```

```
create or replace procedure proc_ex7
is
type tabel is record(titlu painting.painting_title%type,
                     pret buying.price%type);
type tabel_tablouri is table of tabel
index by pls_integer;
```

```
type tabel_c is record(curent art_movement.movement_name%type);
type tabel_curente is table of tabel_c
index by pls_integer;

cursor c is
select a.a_last_name nume_artist, a_first_name prenume_artist
from artist a;

nrCurente number(3);
nrTablouri number(3);
sumaTablouri number(6,2);

tablouri tabel_tablouri;
curente tabel_curente;
nr number(4);

begin
for k in c
loop
dbms_output.new_line();
dbms_output.put_line('-----');
dbms_output.put_line(k.prenume_artist || ' ' || k.nume_artist || ': ');
nrTablouri := 0;
sumaTablouri := 0;

select p.painting_title, b.price
bulk collect into tablouri
from (select pa.painting_title, pa.painting_id, b.price, pa.artist_id
      from painting pa, buying b
     where pa.painting_id = b.painting_id
   order by b.price desc) p, artist a, buying b
  where lower(a.a_last_name) = lower(k.nume_artist) and p.artist_id =
a.artist_id and b.painting_id = p.painting_id and rownum <= 3
  order by b.price desc;

select am.movement_name
bulk collect into curente
from belongs b, artist a, art_movement am
```

```

    where lower(k.numere_artist) = lower(a.a_last_name) and b.artist_id =
a.artist_id

    and am.movement_id = b.movement_id;

nrTablouri := tablouri.count();

if nrTablouri = 0 then
    dbms_output.put_line('Acest artist nu are inregistrat niciun tablou!');

else
    if nrTablouri < 3
        then
            dbms_output.put_line('Acest artist are mai putin de 3 tablouri
inregistrate: ');

    else
        dbms_output.put_line('Top cele mai scumpe tablouri: ');

    end if;

    for i in 1..nrTablouri
    loop
        dbms_output.put_line(i || ':' || tablouri(i).titlu || ' ' ||
tablouri(i).pret || ', mil. $');

        sumaTablouri := sumaTablouri + tablouri(i).pret;
    end loop;

    if nrTablouri >= 3
        then
            dbms_output.put_line('Suma: ' || sumaTablouri || ' mil. $');
            dbms_output.new_line();
    end if;

    end if;

dbms_output.new_line();

nrCurente := curente.count();

if nrCurente = 0 then
    dbms_output.put_line('Acest artist nu a activat in niciun curent!');

else
    dbms_output.put_line('Miscari artistice:');
    for i in 1..nrCurente
    loop
        dbms_output.put_line(curente(i).curent);
    end loop;
    dbms_output.new_line();

```

```

    end if;

    end loop;

end proc_ex7;
/
execute proc_ex7;
/

```

```

188 dbms_output.put_line('Acest artist nu a activat in niciun tablou');
189
190 else
191     dbms_output.put_line('Miscari artistice:');
192     for i in 1..nrCurente
193         loop
194             dbms_output.put_line(curente(i).curent);
195         end loop;
196         dbms_output.new_line();

197     end if;
198
199     end loop;
200 end proc_ex7;
201 /
203
204 execute proc_ex7;
205 /
206

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

homeuser x

Gustav Klimt:
Top cele mai scumpe tablouri:
1: Portrait of Adele Bloch-Bauer I 171,2 mil. \$
2: Portrait of Adele Bloch-Bauer I 171,2 mil. \$
3: Portrait of Adele Bloch-Bauer II 159,8 mil. \$
Suma: 502,2 mil. \$

Miscari artistice:
Symbolism
Art Nouveau
Vienna Secession

Jackson Pollock:
Acest artist sunt mai putin de 3 tablouri inregistrate:
1: No. 5, 1948 177,6 mil. \$

Miscari artistice:
Abstract Expressionism

Vincent van Gogh:
Top cele mai scumpe tablouri:
1: Portrait of Dr. Gachet 161,4 mil. \$

```

100 dbms_output.put_line('Acest artist nu a activat in niciun tablou');
101
102 else
103     dbms_output.put_line('Miscari artistice:');
104     for i in 1..nrCurente
105         loop
106             dbms_output.put_line(curente(i).curent);
107         end loop;
108         dbms_output.new_line();

109     end if;
110
111     end loop;
112 end proc_ex7;
113 /
115
116 execute proc_ex7;
117 /
118

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

homeuser x

Vincent van Gogh:
Top cele mai scumpe tablouri:
1: Portrait of Dr. Gachet 161,4 mil. \$
2: Irises 121,3 mil. \$
3: Starry Night Over the Rhône 101,3 mil. \$
Suma: 384 mil. \$

Miscari artistice:
Post-Impressionism

Thomas Eakins:
Acest artist nu are inregistrat niciun tablou!

Acest artist nu a activat in niciun curent!

Francis Bacon:
Acest artist sunt mai putin de 3 tablouri inregistrate:
1: Triptych, 1976 102,5 mil. \$

Miscari artistice:
Cubism
Abstract Expressionism

```

186      dbms_output.put_line('Acest artist nu a activat in niciun c');
187
188  else
189      dbms_output.put_line('Miscari artistice:');
190      for i in 1..nrCurente
191      loop
192          dbms_output.put_line(curenta(i).current);
193      end loop;
194      dbms_output.new_line();
195
196  end if;
197
198  end loop;
199
200 end proc_ex7;
201 /
203
204 execute proc_ex7;
205 /
206

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Leonardo da Vinci:
Acest artist are mai putin de 3 tablouri inregistrate:
1: Gioconda 860 mil. \$
2: Salvador Mundi 469,7 mil. \$

Miscari artistice:
High Renaissance

Paul Cézanne:
Acest artist are mai putin de 3 tablouri inregistrate:
1: Rudeau, Cruchon et Compotier 92,9 mil. \$

Miscari artistice:
Impressionism
Post-Impressionism

Eugene Henri Paul Gauguin:
Top cele mai scumpe tablouri:
1: The Card Players 284 mil. \$
2: When Will You Marry? 227 mil. \$
3: When Will You Marry? 227 mil. \$
Suma: 738 mil. \$

Miscari artistice:
Primitivism
Post-Impressionism

Pablo Ruiz Picasso:
Top cele mai scumpe tablouri:
1: Women of Algiers 193,5 mil. \$
2: Le Reve 170,1 mil. \$
3: Boy with a Pipe 141 mil. \$
Suma: 504,6 mil. \$

Miscari artistice:
Cubism
Surrealism

Pablo Ruiz Picasso:
Top cele mai scumpe tablouri:
1: Women of Algiers 193,5 mil. \$
2: Le Reve 170,1 mil. \$
3: Boy with a Pipe 141 mil. \$
Suma: 504,6 mil. \$

Miscari artistice:
Cubism
Surrealism

Williem de Kooning:
Top cele mai scumpe tablouri:
1: Interchange 324 mil. \$
2: Woman III 177,6 mil. \$
3: Police Gazette 80,5 mil. \$
Suma: 582,1 mil. \$

Miscari artistice:
Abstract Expressionism

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Se da numele unui vanzator și numele unei expoziții. Dorim să aflăm cat comision a luat casa de vanzari prin care a fost vândut un tablou de către acel vanzator în cadrul acelei expoziții. Atentie! Unele tablouri au fost vândute prin achiziții private, însemnând ca nu este menționată casa de vanzare, iar altele au fost vândute de mai multe ori.

```
--ca parametrii vom primi numele sellerului și vom returna comisionul
--excepțiile apar atunci cand un seller a vândut mai multe tablouri la aceeași expoziție
--expozitie prin aceeași casa de vânzare, sau dacă nu a vândut niciunul

create or replace function function_ex8
    (nume_seller seller.s_last_name%type default 'Gioconda',
     expoziție exhibition.exhibition_title%type default 'Shape Lab')
return number is
    comision auction_house.commission_pct%type;
    idSeller number;
begin
    select s.seller_id
    into idSeller
    from seller s
    where lower(s.s_last_name) = lower(nume_seller);

    select a.commission_pct
    into comision
    from buying b, auction_house a, exhibition e
    where b.seller_id = idSeller and b.exhibition_id = e.exhibition_id
    and lower(e.exhibition_title) = lower(expozitie)
    and b.house_id = a.house_id;
    return comision;

exception
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000,
```

```
'Nu exista casa de vanzari.';

WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20001,
'Exista prea multe case de vanzari.';

WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end function_ex8;
/
begin
dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'Building
Collections'));
end;
/
```

Pentru Altmann si expozitia Building Collections care are id-ul 4, exista doar un tablou vandut prin casa cu id-ul 3. Aceasta casa este Sotheby si are comisionul 4.8.

The screenshot shows the Oracle SQL Developer interface. At the top, there is a code editor window with the following PL/SQL code:

```
335 begin
336   dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'Building Collections'));
337 end;
338 /
```

Below the code editor is a toolbar with icons for Script Output, Query Result, and other database operations. A status bar at the bottom indicates "Task completed in 0,06 seconds".

In the main workspace, the message "PL/SQL procedure successfully completed." is displayed.

At the bottom, there is another "Dbms Output" window showing the result of the procedure execution:

```
Dbms Output
+ - Buffer Size: 20000
homeuser
Numarul este 4,8
```

```
begin
dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'In the
Making'));
end;
/
```

Pentru Altmann si expozitia In the Making care are id-ul 2, exista 2 vanzari de tablouri.

```

341 begin
342     dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'In the Making'));
343 end;

```

Script Output | Query Result | Task completed in 0,087 seconds
Error report -
ORA-20001: Există prea multe case de vanzari.
ORA-06512: la "HOMEUSER.FUNCTION_EX8", linia 32
ORA-06512: la linia 2

```

begin
    dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'Shape Lab'));
end;
/

```

Pentru Altmann și expoziția Shape Lab care are id-ul 3, casa de vanzari nu este menționată.

```

346 begin
347     dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'Shape Lab'));
348 end;
349 /

```

Script Output | Query Result | Task completed in 0,073 seconds
dbms_output.put_line('Numarul este'|| function_ex8('Altmann', 'Shape Lab'));
end;
Error report -
ORA-20000: Nu există casa de vanzari.
ORA-06512: la "HOMEUSER.FUNCTION_EX8", linia 28
ORA-06512: la linia 2

9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru a incuraja achizițiile, se organizează o tombola, iar unul dintre cumpăratorii va primi un premiu surpriză. Primul cumpărator (în ordine alfabetica) care locuiește în USA și care a cumpărat cel puțin un tablou mai vechi de 1950 prin intermediul acelei case de vanzari data ca parametru, este ales castigator. Afisați acel nume. Dacă niciun cumpărator nu intrunește aceste condiții, premiul nu va fi acordat.

```

--ca parametru de intrare primim numele casei, ca parametru de ieșire, numele
--castigatorului (cumpărator)

--excepție poate apărea atunci cand niciun cumpărat nu respectă cerințele
--excepție de too many rows nu o să apara deoarece avem un order by și il luăm pe
--primul în ordine alfabetica

```

```
create or replace PROCEDURE proc_ex9
  (casa in auction_house.house_name%type,
   nume out buyer.b_last_name%type)
IS
BEGIN

  select bb.b_last_name
  into nume
  from (select b.b_last_name, b.buyer_id
         from buyer b, address a, buying bu, painting p, auction_house ah
        where a.country = 'USA' and p.painting_year < 1950
          and b.buyer_id = bu.buyer_id and a.address_id = b.address_id
          and p.painting_id = bu.painting_id and ah.house_id = bu.house_id
          and ah.house_name = casa
         order by b.b_last_name) p, buyer bb
  where bb.buyer_id = p.buyer_id and rownum < 2
  order by p.b_last_name;

exception
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,
      'Nu exista inregistrari care sa respecte cerintele.');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END proc_ex9;
/
```

DECLARE

```
  acestNume buyer.b_last_name%type;
```

```
BEGIN
  proc_ex9('Sotheby', acestNume);
  dbms_output.put_line(acestNume);
```

```

END;
/

--Daca analizam baza de date, vom observa ca:
--14, 13, 4, 10, 5, 16, 17, 18 sunt id-urile adreselor din USA
--staechelin, griffin, james, wynn, whitney locuiesc in USA (cu id-urile
2,6,8,9,10)
--In cazul de mai jos, am dat 'Sotheby', casa de vanzari cu id-ul 3.
--Dintre cei 5, cei care au cumparat vreun tablou prin casa 3:
--2, 9, 6, 10 (staechelin, griffin, wynn, whitney)
--id-urile tablourilor cumpарате de ei: 15, 3, 10, 27.
--dintre aceste tablouri, cel cu id-ul 27 este din 1976, deci whitney este exclus
--dintre cei ramasi, primul in ordine alfabetica este Griffin.

```

```

307 | DECLARE
308 |     acestNume buyer.b_last_name%type;
309 |
310 | BEGIN
311 |     proc_ex9('Sotheby', acestNume);
312 |     dbms_output.put_line('Castigatorul este ' || acestNume);
313 | END;
314 |
315 |

```

Script Output x | Query Result x | Query Result 1 x
Task completed in 0,032 seconds

PL/SQL procedure successfully completed.

Dbms Output
+ | Buffer Size: 20000
homeuser x

Castigatorul este Griffin

```

DECLARE
    acestNume buyer.b_last_name%type;

BEGIN
    proc_ex9('Christie', acestNume);
    dbms_output.put_line(acestNume);

END;
/

```

```

317 =DECLARE
318     acestNume buyer.b_last_name%type;
319
320 BEGIN
321     proc_ex9('Christie', acestNume);
322     dbms_output.put_line('Castigatorul este ' || acestNume);
323 END;
324 /

```

ORA-20000: Nu există înregistrări care să respecte cerințele.
ORA-06512: la "HOMEUSER.PROC_EX9", linia 22
ORA-06512: la linia 5
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error' was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

De regula, expozițiile se tin doar vineri, sambata și duminica, începând cu orele 17:00.

In acest interval, nu sunt posibile modificări aduse acestui tabel, în schimb, în restul timpului, modificările sunt permise. Declarati un trigger care să interzică adăugarea, stergerea sau updatearea datelor în tabelul exhibition în acest interval orar.

```
--vineri este ziua 5, sambata 6 si duminica 7
--intre 8 si 17 se pot adauga expoziții, in schimb, dupa 17, nu se mai pot.
```

```

CREATE OR REPLACE TRIGGER trig_exhibition
    BEFORE INSERT OR DELETE OR UPDATE on exhibition
BEGIN
    IF (TO_CHAR(SYSDATE,'D') BETWEEN 5 and 7)
        AND (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 17)

    THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20001,'Inserarea în tabelul exhibition
este permisa în afara orelor de expoziție!');

        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20002,'Stergerea din tabelul exhibition
este permisa în afara orelor de expoziție!');

        ELSE
            RAISE_APPLICATION_ERROR(-20003,'Actualizările în tabelul exhibition

```

```
sunt permise inafara orelor de expozitie!');

END IF;

END;
```

```
872 begin
873 dbms_output.put_line(TO_CHAR(SYSDATE, 'D'));
874 end;
875

PL/SQL procedure successfully completed.
```

```
insert into exhibition(exhibition_id, exhibition_title, exhibition_date,
exhibition_city) values
(7, 'Sun is up', to_date('04-10-2020', 'dd-mm-yyyy'), 'Los Angeles');
```

```
361 RAISE_APPLICATION_ERROR(-20002,'Stergerea din tabelul exhibition
este permisa inafara orelor de expozitie!');

362
363
364 ELSE
365 RAISE_APPLICATION_ERROR(-20003,'Actualizările în tabelul exhibition
sunt permise inafara orelor de expozitie!');

366 END IF;
367
368 END IF;
369 END;
370
371 /
372
373 insert into exhibition(exhibition_id, exhibition_title, exhibition_date, exhibition_city) value
374 (7, 'Sun is up', to_date('04-10-2020', 'dd-mm-yyyy'), 'Los Angeles');
```

```
delete from exhibition
where exhibition_id = 6;
```

```

375
376 delete from exhibition
377 where exhibition_id = 6;
378
379
380

```

Script Output | Query Result | Task completed in 0.032 seconds

Error starting at line : 376 in command -
 delete from exhibition
 where exhibition_id = 6
 Error report -
 ORA-20002: Stergerea din tabelul exhibition
 este permisa inafara orelor de expoziție!
 ORA-06512: la "HOMEUSER.TRIG_EXHIBITION", linia 11
 ORA-04088: eroare in timpul execuției triggerului 'HOMEUSER.TRIG_EXHIBITION'

```

update exhibition
set exhibition_title = 'Sun'
where exhibition_id = 6;

```

```

378
379 update exhibition
380 set exhibition_title = 'Sun'
381 where exhibition_id = 6;
382
383
384

```

Script Output | Query Result | Task completed in 0.042 seconds

Error starting at line : 379 in command -
 update exhibition
 set exhibition_title = 'Sun'
 where exhibition_id = 6
 Error report -
 ORA-20003: Actualizările în tabelul exhibition
 sunt permise inafara orelor de expoziție!
 ORA-06512: la "HOMEUSER.TRIG_EXHIBITION", linia 15
 ORA-04088: eroare in timpul execuției triggerului 'HOMEUSER.TRIG_EXHIBITION'

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

In momentul in care se face un update pentru pretul cu care a fost vandut un anumit tablou, pentru a evita posibile greseli, vom folosi un trigger care sa nu permita ca pretul cel mai mic sa devina mai mare decat media preturilor deja existente, pretul cel mai mare sa nu devina mai mic decat media preturilor si totodata, pretul cel mai mare sa nu devina mai mare decat suma tuturor preturilor din baza de date.

```

--am folosit un pachet in care retineam minimul, maximul, media si suma preturilor
--avem un trigger la nivel de comanda in care vom selecta aceste informatii in
--variabilele din pachet. Triggerul la nivel de linie se va raporta la valorile

```

--acelor variabile.

create or replace package pachet_trig

is

 minp buying.price%type;

 avp buying.price%type;

 maxp buying.price%type;

 sump buying.price%type;

end pachet_trig;

/

create or replace trigger trigg11_first

before update of price on buying

begin

 select min(b.price), max(b.price), avg(b.price), sum(b.price)

 into pachet_trig.minp, pachet_trig.maxp, pachet_trig.avp, pachet_trig.sump

 from buying b;

end;

/

create or replace trigger trigg11_plus

before update of price on buying

for each row

begin

 if (:OLD.price = pachet_trig.minp) and (:NEW.price > pachet_trig.avp)

 then

 RAISE_APPLICATION_ERROR(-20001, 'Pretul minim nu poate fi mai mare decat media preturilor!');

 elsif (:OLD.price = pachet_trig.maxp) and (:NEW.price < pachet_trig.avp)

 then

 RAISE_APPLICATION_ERROR(-20001, 'Pretul maxim nu poate fi mai mic decat media preturilor!');

 elsif (:OLD.price = pachet_trig.maxp) and (:NEW.price > pachet_trig.sump)

 then

```
RAISE_APPLICATION_ERROR(-20001, 'Pretul maxim nu poate fi mai mare decat suma  
celorlalte tablouri!');
```

```
end if;  
end;  
  
/  
  
855 begin  
856   dbms_output.put_line(pachet_trig.minp);  
857   dbms_output.put_line(pachet_trig.maxp);  
858   dbms_output.put_line(pachet_trig.avp);  
859   dbms_output.put_line(pachet_trig.sump);  
860 end;  
861 /  
862
```

PL/SQL procedure successfully completed.

```
Dbms Output  
+ - homeuser | Buffer Size: 20000 |  
homeuser x  
71,5  
860  
189,3  
5678,9
```

```
update buying  
set price = 190  
where price = (select min(price) from buying);
```

```
863 update buying  
864 set price = 190  
865 where price = (select min(price) from buying);  
866  
  
Dbms Output  
+ - homeuser | Buffer Size: 20000 |  
homeuser x  
where price = (select min(price) from buying)  
Error report -  
ORA-20001: Pretul minim nu poate fi mai mare decat media preturilor!  
ORA-06512: la "HOMEUSER.TRIGG11_PLUS", linia 6  
ORA-04088: eroare in timpul executiei triggerului 'HOMEUSER.TRIGG11_PLUS'
```

```
update buying  
set price = 6000  
where price = (select max(price) from buying);
```

```

867| update buying
868| set price = 600
869| where price = (select max(price) from buying);
870| 
871| update buying
872| set price = 180
873| where price = (select max(price) from buying);
874| 
```

Script Output | Query Result | Task completed in 0,061 seconds
where price = (select max(price) from buying)
Error report -
ORA-20001: Pretul maxim nu poate fi mai mare decat suma celorlalte tablouri!
ORA-06512: la "HOMEUSER.TRIGG11_PLUS", linia 14
ORA-04088: eroare in timpul executiei triggerului 'HOMEUSER.TRIGG11_PLUS'

update buying
set price = 180
where price = (select max(price) from buying);

871| update buying
872| set price = 180
873| where price = (select max(price) from buying);
874|

Script Output | Query Result | Task completed in 0,055 seconds
set price = 180
where price = (select max(price) from buying)
Error report -
ORA-20001: Pretul maxim nu poate fi mai mic decat media preturilor!
ORA-06512: la "HOMEUSER.TRIGG11_PLUS", linia 10
ORA-04088: eroare in timpul executiei triggerului 'HOMEUSER.TRIGG11_PLUS'

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

Ne dorim un trigger care sa se declanseze in momentul efectuarii de comenzi LDD asupra obiectelor din schema personală. Vrem sa retinem numele bazei de date, userul, evenimentul efectuat, tipul si numele obiectului referit.

```

CREATE TABLE a_user
(
    nume_baza VARCHAR2(50),
    user_logat VARCHAR2(40),
    eveniment VARCHAR2(30),
    tip_object_referit VARCHAR2(30),
    nume_object_referit VARCHAR2(30),
    data TIMESTAMP(3));

```

HOMEUSER.A_USER	
NUME_BAZA	VARCHAR2 (50 BYTE)
USER_LOGAT	VARCHAR2 (40 BYTE)
EVENIMENT	VARCHAR2 (30 BYTE)
TIP_OBIECT_REFERIT	VARCHAR2 (30 BYTE)
NUME_OBJECT_REFERIT	VARCHAR2 (30 BYTE)
DATA	TIMESTAMP (3)

```

CREATE OR REPLACE TRIGGER a_schema
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO a_user
        VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER,
        SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE,
        SYS.DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));
END;
/
CREATE TABLE tabel (coloana_1 number(2));
ALTER TABLE tabel ADD (coloana_2 number(2));

INSERT INTO tabel VALUES (1,2);
CREATE INDEX ind_tabel ON tabel(coloana_1);

SELECT * FROM a_user;

```

416 |

417 | **SELECT * FROM a_user;**

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for Script Output, Query Result, and three other tabs labeled 1, 2, and 3. Below the tabs, a message says "All Rows Fetched: 15 in 0.011 seconds". The main area displays a table with 15 rows of data. The columns are labeled: #, NUME_BAZA, USER_LOGAT, EVENIMENT, TIP_OBIECT_REFERIT, NUME_OBJECT_REFERIT, and DATA. The data rows show various system events and object creation details.

#	NUME_BAZA	USER_LOGAT	EVENIMENT	TIP_OBIECT_REFERIT	NUME_OBJECT_REFERIT	DATA
1	XE	HOMEUSER CREATE PROCEDURE	PROC TRIG SECOL	07-01-2021 20:01:13	,0900000000	
2	XE	HOMEUSER CREATE TRIGGER	TRIG SECOL	07-01-2021 20:01:13	,1070000000	
3	XE	HOMEUSER CREATE TRIGGER	TRIG EXHIBITION	07-01-2021 20:01:19	,8740000000	
4	XE	HOMEUSER CREATE PACKAGE	PACHET GALLERY	07-01-2021 20:27:45	,3590000000	
5	XE	HOMEUSER CREATE PACKAGE BODY	PACHET GALLERY	07-01-2021 20:27:56	,3030000000	
6	XE	HOMEUSER CREATE PACKAGE BODY	PACHET GALLERY	07-01-2021 20:28:59	,3450000000	
7	XE	HOMEUSER CREATE PACKAGE	PACHET GALLERY	07-01-2021 20:36:11	,1820000000	
8	XE	HOMEUSER CREATE PACKAGE BODY	PACHET GALLERY	07-01-2021 20:36:41	,8260000000	
9	XE	HOMEUSER CREATE PACKAGE	PACHET GALLERY	07-01-2021 20:42:52	,9740000000	
10	XE	HOMEUSER CREATE PACKAGE BODY	PACHET GALLERY	07-01-2021 20:43:07	,2120000000	
11	XE	HOMEUSER CREATE PACKAGE	PACHET GALLERY	07-01-2021 20:45:57	,8960000000	
12	XE	HOMEUSER CREATE PACKAGE BODY	PACHET GALLERY	07-01-2021 20:46:14	,7380000000	
13	XE	HOMEUSER CREATE PROCEDURE	PROC EX6	07-01-2021 21:22:29	,2320000000	
14	XE	HOMEUSER CREATE PROCEDURE	PROC EX7	07-01-2021 21:24:04	,1400000000	
15	XE	HOMEUSER CREATE FUNCTION	FUNC EX8	07-01-2021 21:25:53	,0360000000	

```
SELECT * FROM tabel;
```

```

388 | SELECT * FROM tabel;
389 |

```

COLOANA_1	COLOANA_2
1	2
2	2

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului:

Ne dorim un pachet care sa contina toate obiectele create anterior.

```

CREATE OR REPLACE PACKAGE pachet_gallery AS

procedure proc_ex6;
procedure proc_ex7;

PROCEDURE proc_ex9
(casa in auction_house.house_name%type,
nume out buyer.b_last_name%type);

function function_ex8
(numere_seller seller.s_last_name%type default 'Gioconda',
expozitie exhibition.exhibition_title%type default 'Shape Lab')
return number;

END pachet_gallery;
/

CREATE OR REPLACE PACKAGE BODY pachet_gallery AS

function function_ex8
(numere_seller seller.s_last_name%type default 'Gioconda',
expozitie exhibition.exhibition_title%type default 'Shape Lab')
return number is
comision auction_house.commission_pct%type;
idSeller number;
begin

```

```

select s.seller_id
into idSeller
from seller s
where lower(s.s_last_name) = lower(numere_seller);

select a.commission_pct
into comision
from buying b, auction_house a, exhibition e
where b.seller_id = idSeller and b.exhibition_id = e.exhibition_id
and lower(e.exhibition_title) = lower(expozitie)
and b.house_id = a.house_id;
return comision;

exception
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20000,
'Nu exista casa de vanzari.');

WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20001,
'Există prea multe case de vanzari.');

WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

end function_ex8;

procedure proc_ex6
is
type tabells is record(titlu exhibition.exhibition_title%type,
nrTab number(2),
nrTabV number (2),
rap number(5,2));
type vector_tabl is varray(20) of tabells;
t vector_tabl := vector_tabl();

```

```
nr number(2);
r number(5,2);

begin
    select e.exhibition_title, count(*), 0, 0
    bulk collect into t
    from exhibition e, participate p
    where p.exhibition_id = e.exhibition_id
    group by e.exhibition_title;

    for i in t.first..t.last
    loop
        select count(b.exhibition_id)
        into nr
        from exhibition e, buying b
        where e.exhibition_id = b.exhibition_id
        and e.exhibition_title = t(i).titlu
        group by e.exhibition_title;

        t(i).nrTabV := nr;
        r := t(i).nrTabV * 100 / t(i).nrTab;
        t(i).rap := r;
    end loop;

    for i in 1..t.last
    loop
        dbms_output.put_line('Titlu: ' || t(i).titlu);
        dbms_output.put_line('Nr de tablouri ' || ' ' || t(i).nrTab);
        dbms_output.put_line('Nr de tablouri vandute ' || ' ' || t(i).nrTabV);
        dbms_output.put_line('Au fost vandute ' || t(i).rap || '% din tablourile expuse.');

        dbms_output.new_line();

        update exhibition_ado
        set nr_tablouri = t(i).nrTab
        where exhibition_title = t(i).titlu;
```

```
update exhibition_ado
set nr_tablouri_vandute = t(i).nrTabV
where exhibition_title = t(i).titlu;

update exhibition_ado
set raport_tab = t(i).rap
where exhibition_title = t(i).titlu;

end loop;

end proc_ex6;

procedure proc_ex7
is
type tabel is record(titlu painting.painting_title%type,
                     pret buying.price%type);
type tabel_tablouri is table of tabel
index by pls_integer;

type tabel_c is record(curent art_movement.movement_name%type);
type tabel_curente is table of tabel_c
index by pls_integer;

cursor c is
select a.a_last_name nume_artist, a_first_name prenume_artist
from artist a;

nrCurente number(3);
nrTablouri number(3);
sumaTablouri number(6,2);

tablouri tabel_tablouri;
curente tabel_curente;
```

```
nr number(4);

begin

    for k in c
    loop

        dbms_output.new_line();
        dbms_output.put_line('-----');
        dbms_output.put_line(k.prenume_artist || ' ' || k.nume_artist || ': ');
        nrTablouri := 0;
        sumaTablouri := 0;

        select p.painting_title, b.price
        bulk collect into tablouri
        from (select pa.painting_title, pa.painting_id, b.price, pa.artist_id
               from painting pa, buying b
              where pa.painting_id = b.painting_id
              order by b.price desc) p, artist a, buying b
        where lower(a.a_last_name) = lower(k.nume_artist) and p.artist_id =
a.artist_id and b.painting_id = p.painting_id and rownum <= 3
        order by b.price desc;

        select am.movement_name
        bulk collect into curente
        from belongs b, artist a, art_movement am
        where lower(k.nume_artist) = lower(a.a_last_name) and b.artist_id =
a.artist_id
        and am.movement_id = b.movement_id;

        nrTablouri := tablouri.count();

        if nrTablouri = 0 then
            dbms_output.put_line('Acest artist nu are inregistrat niciun tablou!');
        else
            if nrTablouri < 3
            then
                dbms_output.put_line('Acest artist are mai putin de 3 tablouri
inregistrate: '');
```

```

        else
            dbms_output.put_line('Top cele mai scumpe tablouri: ');

        end if;
        for i in 1..nrTablouri
        loop
            dbms_output.put_line(i || ':' || tablouri(i).titlu || ' ' ||
tablouri(i).pret || ' mil. $');
            sumaTablouri := sumaTablouri + tablouri(i).pret;
        end loop;
        if nrTablouri >= 3
        then
            dbms_output.put_line('Suma: ' || sumaTablouri || ' mil. $');
            dbms_output.new_line();
        end if;
        end if;

        dbms_output.new_line();

        nrCurente := curente.count();
        if nrCurente = 0 then
            dbms_output.put_line('Acest artist nu a activat in niciun curent!');
        else
            dbms_output.put_line('Miscari artistice:');
            for i in 1..nrCurente
            loop
                dbms_output.put_line(curente(i).curent);
            end loop;
            dbms_output.new_line();
        end if;

        end loop;
    end proc_ex7;

```

```
PROCEDURE proc_ex9
  (casa in auction_house.house_name%type,
   nume out buyer.b_last_name%type)
IS
BEGIN

  select bb.b_last_name
  into nume
  from (select b.b_last_name, b.buyer_id
         from buyer b, address a, buying bu, painting p, auction_house ah
        where a.country = 'USA' and p.painting_year < 1950
          and b.buyer_id = bu.buyer_id and a.address_id = b.address_id
          and p.painting_id = bu.painting_id and ah.house_id = bu.house_id
          and ah.house_name = casa
        order by b.b_last_name) p, buyer bb
  where bb.buyer_id = p.buyer_id and rownum < 2
  order by p.b_last_name;

  exception
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20000,
      'Nu exista inregistrari care sa respecte cerintele.');
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END proc_ex9;

END pachet_gallery;
/
```

```
DECLARE
  acestNume buyer.b_last_name%type;
BEGIN
```

```

pachet_gallery.proc_ex6();
pachet_gallery.proc_ex7();
dbms_output.put_line('Comisionul este '|| pachet_gallery.func_ex8 ('Altmann',
'Building Collections'));
pachet_gallery.proc_ex9('Sotheby', acestNume);
dbms_output.put_line('Castigatorul este ' || acestNume);

END;
/

```

```

Worksheet  Query Builder
653 END proc_ex9;
654
655 END pachet_gallery;
656 /
657
658=DECLARE
659   acestNume buyer.b_last_name%type;
660   BEGIN
661     pachet_gallery.proc_ex6();
662     pachet_gallery.proc_ex7();
663     dbms_output.put_line('Comisionul este '|| pachet_gallery.func_ex8("Irises"));
664     pachet_gallery.proc_ex9('Sotheby', acestNume);
665     dbms_output.put_line('Castigatorul este ' || acestNume);
666   END;
667 /
668
669
670

Script Output  X  Query Result  X  Query Result 1  X  Query Result 2  X  Query Result 3  X
Package PACHET_GALLERY compiled
Package Body PACHET_GALLERY compiled

----- Pablo Ruiz Picasso: -----
Top cele mai scumpe tablouri:
1: Women of Algiers 193,5 mil. $
2: Le Reve 170,1 mil. $
3: Boy with a Pipe 141 mil. $
Suma: 504,6 mil. $

----- Miscari artistice: -----
Cubism
Surrealism

----- Williem de Kooning: -----
Top cele mai scumpe tablouri:
1: Interchange 324 mil. $
2: Woman III 177,6 mil. $
3: Police Gazette 80,5 mil. $
Suma: 582,1 mil. $

----- Miscari artistice: -----
Abstract Expressionism

----- Comisionul este 3,9
----- Castigatorul este Griffin

```

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.

Ne dorim sa afisam pentru fiecare artist, care sunt tablourile cu care acesta figureaza, folosind un tip de date complex.

Pe de alta parte, vrem sa afisam pentru fiecare vanzator, ce tablouri a vandut si carui artist ii apartine fiecare dintre tablourile lui, utilizand tipul de date complex de mai devreme.

Totodata, vrem sa stim si care este cel mai scump tablou vandut de catre fiecare vanzator.

Pentru a rezolva exercitiul, vom folosi un pachet in care vom avea doua tipuri de date complexe. Un vector de vectori pentru a retine tablourile artistilor si un vector de tablouri imbricate pentru a retine tablourile vandute de vanzatori.

In pachet vom mai avea:

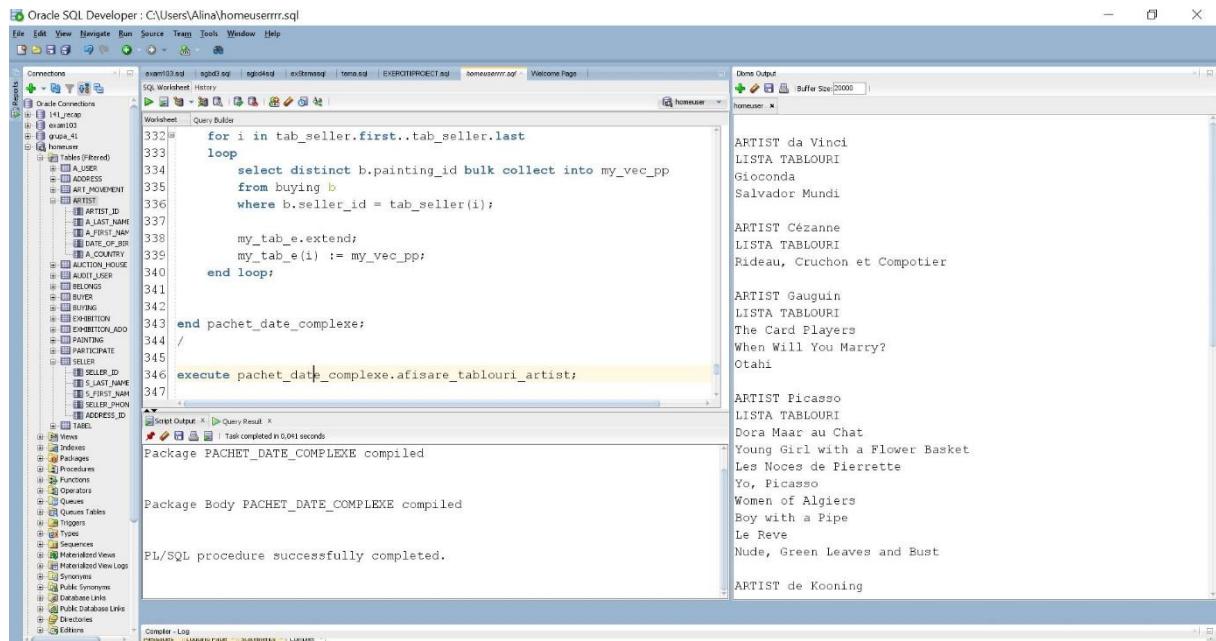
- o procedura afisare_tablouri_artist care va afisa tablourile fiecarui artist,

- o procedura afisare_tablouri_seller care doar afiseaza tablourile vandute de vanzatori

- o procedura afisare_nume_artist care primeste id-ul unui tablou si ofera ca rezultat numele artistului caruia ii apartine

- o procedura *celMaiScump* care primește id-ul vanzatorului și oferă ca rezultat titlul celui mai scump tablou vândut de acesta

- o procedura afisare_tablouri_seller_artist care afiseaza pentru fiecare vanzator tablourile vandute si cui ii apartine fiecare tablou, in plus, care este cel mai scump tablou vandut de catre fiecare vanzator. Aceasta procedura foloseste ambele tipuri de date complexe.



```
create or replace package pachet_date_complexe as  
    type vec_artists is varray(1000) of artist.artist_id%type;  
    my_vec a vec_artists;
```

```
type vec_paintings is varray(1000) of painting.painting_id%type;  
type tabl is table of vec_paintings;  
my_tab tabl := tabl();  
my_vec p vec_paintings;
```

```
type type_tab_seller is table of seller.seller_id%type;  
tab seller type tab seller;
```

```
type tab_e is table of vec_paintings;  
my_tab_e tab_e := tab_e();  
my vec pp vec paintings;
```

```
procedure afisare_tablouri_artist;
procedure afisare_tablouri_seller;
procedure afisare_tablouri_seller_artist;

procedure afisare_nume_artist(idTablou IN painting.painting_id%TYPE,
numeArtist OUT artist.a_last_name%type);

procedure celMaiScump(idSeller IN seller.seller_id%TYPE,
titluTab OUT painting.painting_title%type);

end pachet_date_complex;
/

create or replace package body pachet_date_complex as

procedure afisare_tablouri_artist is
    n artist.a_last_name%type;
    n1 artist.a_last_name%type;
    idul artist.artist_id%type;
    titlul painting.painting_title%type;
    titlull painting.painting_title%type;
    idultablou painting.painting_id%type;

    cursor c1 is
        select a.a_last_name numeArtist, a.artist_id idArtist
        from artist a;

    cursor c is
        select p.painting_title numetablou, p.painting_id idTablou
        from painting p;

begin

    for i in my_tab.first..my_tab.last loop
        dbms_output.new_line();

```

```
open c1;
loop
    fetch c1 into n1, idul;
    EXIT WHEN c1%NOTFOUND;

    if idul = my_vec_a(i)
        then n := n1;
    end if;
end loop;
close c1;

dbms_output.put_line('ARTIST ' || n);
if my_tab(i).count = 0 then
    dbms_output.put_line('NU SUNT TABLOURI');
else
    dbms_output.put_line('LISTA TABLOURI');
    for j in my_tab(i).first..my_tab(i).last
    loop
        open c;
        loop
            fetch c into titlull, idultablou;
            EXIT WHEN c%NOTFOUND;

            if idultablou = my_tab(i)(j)
                then titlul := titlull;
            end if;
        end loop;
        close c;

        dbms_output.put_line(titlul);
    end loop;
end if;
end loop;

end afisare_tablouri_artist;

procedure afisare_tablouri_seller is
```

```
n seller.s_last_name%type;
n1 seller.s_last_name%type;
idul seller.seller_id%type;
titlul painting.painting_title%type;
titlull painting.painting_title%type;
idulTablou painting.painting_id%type;

cursor c2 is
select s.s_last_name numeSeller, s.seller_id idSeller
from seller s;

cursor c is
select p.painting_title numeTablou, p.painting_id idTablou
from painting p;

begin
for i in my_tab_e.first..my_tab_e.last
loop
dbms_output.new_line();

open c2;
loop
fetch c2 into n1, idul;
EXIT WHEN c2%NOTFOUND;

if idul = tab_seller(i)
then n := n1;
end if;

end loop;
close c2;

dbms_output.put_line('Seller ' || n);
if my_tab_e(i).count = 0 then
dbms_output.put_line('NU SUNT TABLOURI');
else
dbms_output.put_line('LISTA TABLOURI');


```

```
for j in my_tab_e(i).first..my_tab_e(i).last
loop
    open c;
    loop
        fetch c into titlull, idulTablou;
        EXIT WHEN c%NOTFOUND;

        if idulTablou = my_tab_e(i)(j)
        then titlul := titlull;
        end if;
        end loop;
        close c;

        dbms_output.put_line(titlul);
    end loop;
end if;
end loop;

end afisare_tablouri_seller;

procedure afisare_nume_artist(idTablou IN painting.painting_id%TYPE,
                               numeArtist OUT artist.a_last_name%type) is

    n artist.a_last_name%type;
    idul artist.artist_id%type;
    idArtist artist.artist_id%type;
    iterator number;

    cursor c1 is
        select a.a_last_name numeArtist, a.artist_id idArtist
        from artist a;

begin
    for i in my_tab.first..my_tab.last
    loop
        if my_tab(i).count != 0
        then
```

```

        for j in my_tab(i).first..my_tab(i).last
        loop
            if idTablou = my_tab(i)(j)
                then iterator := i;
            end if;
        end loop;
    end if;

    end loop;

    idArtist := my_vec_a(iterator);

    open c1;
    loop
        fetch c1 into n, idul;
        EXIT WHEN c1%NOTFOUND;

        if idul = idArtist
            then numeArtist := n;
        end if;
    end loop;
    close c1;

end afisare_nume_artist;

procedure celMaiScump(idSeller IN seller.seller_id%TYPE,
titluTab OUT painting.painting_title%type) is

    idul seller.seller_id%type;
    idTablou painting.painting_id%type;
    idulTablou painting.painting_id%type;
    titlull painting.painting_title%type;
    iterator number;
    iterator1 number;

    cursor c2 is
        select s.s_last_name numeSeller, s.seller_id idSeller

```

```
from seller s;

cursor c is
select p.painting_title numetabou, p.painting_id idTablou
from painting p;

begin

select b.painting_id into idTablou
from buying b
where b.seller_id = idSeller
and b.price = (select max(b.price)
                from buying b
                where b.seller_id = idSeller);

for i in my_tab_e.first..my_tab_e.last
loop

if idSeller = tab_seller(i) and my_tab(i).count != 0
then
    open c;
    loop
        fetch c into titlull, idulTablou;
        EXIT WHEN c%NOTFOUND;

        if idulTablou = idTablou
        then titluTab := titlull;
        end if;
    end loop;
    end if;
end loop;

end celMaiScump;

procedure afisare_tablouri_seller_artist is

n seller.s_last_name%type;
```

```
n1 seller.s_last_name%type;
idul seller.seller_id%type;
idulSellerului seller.seller_id%type;
titlul painting.painting_title%type;
titlull painting.painting_title%type;
titlulScump painting.painting_title%type;
idulTablou painting.painting_id%type;
idulArtist artist.a_last_name%type;
artistul artist.artist_id%type;
numeArtist artist.a_last_name%type;
numele artist.a_last_name%type;

cursor c is
select p.painting_title numeTablou, p.painting_id idTablou
from painting p;

cursor c2 is
select s.s_last_name numeSeller, s.seller_id idSeller
from seller s;

begin

for i in my_tab_e.first..my_tab_e.last loop
dbms_output.new_line();
open c2;
loop
fetch c2 into n1, idul;
EXIT WHEN c2%NOTFOUND;

if idul = tab_seller(i)
then n := n1;
end if;
end loop;
close c2;

dbms_output.new_line();
dbms_output.put_line('-----');
```

```

dbms_output.put_line('Seller ' || n);

if my_tab_e(i).count = 0 then
    dbms_output.put_line('NU SUNT TABLOURI');
else

    dbms_output.put_line('LISTA TABLOURI');
    for j in my_tab_e(i).first..my_tab_e(i).last
    loop
        open c;
        loop
            fetch c into titlull, idulTablou;
            EXIT WHEN c%NOTFOUND;

            if idulTablou = my_tab_e(i)(j)
            then
                titlul := titlull;
                numele := '';
                afisare_nume_artist(idulTablou, numele);
            end if;
            end loop;
            close c;

            dbms_output.put_line(titlul || ' - ' || numele);
        end loop;

        titlulScump := '';
        celMaiScump(tab_seller(i), titlulScump);
        dbms_output.new_line();
        dbms_output.put_line('Cel mai scump tablou vandut: ' || titlulScump);
    end if;
end loop;

end afisare_tablouri_seller_artist;

begin

```

```
select a.artist_id bulk collect into my_vec_a
from artist a;

for i in my_vec_a.first..my_vec_a.last
loop
    select p.painting_id bulk collect into my_vec_p
    from painting p
    where p.artist_id = my_vec_a(i);

    my_tab.extend;
    my_tab(i) := my_vec_p;
end loop;

-- 

select s.seller_id bulk collect into tab_seller
from seller s;

for i in tab_seller.first..tab_seller.last
loop
    select distinct b.painting_id bulk collect into my_vec_pp
    from buying b
    where b.seller_id = tab_seller(i);

    my_tab_e.extend;
    my_tab_e(i) := my_vec_pp;
end loop;

end pachet_date_complex;
/

execute pachet_date_complex.afisare_tablouri_artist;
```

```

332    for i in tab_seller.first..tab_seller.last
333    loop
334        select distinct b.painting_id bulk collect into my_vec_pp
335        from buying b
336        where b.seller_id = tab_seller(i);
337
338        my_tab_e.extend;
339        my_tab_e(i) := my_vec_pp;
340    end loop;
341
342 end pachet_date_complexe;
344 /
345
346 execute pachet_date_complexe.afisare_tablouri_artist;
347

```

Script Output: Task completed in 0,041 seconds

Package PACHET_DATE_COMPLEXE compiled

Package Body PACHET_DATE_COMPLEXE compiled

PL/SQL procedure successfully completed.

ARTIST da Vinci
LISTA TABLOURI
Gioconda
Salvador Mundi

ARTIST Cézanne
LISTA TABLOURI
Rideau, Cruchon et Compotier

ARTIST Gauguin
LISTA TABLOURI
The Card Players
When Will You Marry?
Otahai

ARTIST Picasso
LISTA TABLOURI
Dora Maar au Chat
Young Girl with a Flower Basket
Les Noces de Pierrette
Yo, Picasso
Women of Algiers
Boy with a Pipe
Le Reve
Nude, Green Leaves and Bust

ARTIST de Kooning

```

332    for i in tab_seller.first..tab_seller.last
333    loop
334        select distinct b.painting_id bulk collect into my_vec_pp
335        from buying b
336        where b.seller_id = tab_seller(i);
337
338        my_tab_e.extend;
339        my_tab_e(i) := my_vec_pp;
340    end loop;
341
342 end pachet_date_complexe;
344 /
345
346 execute pachet_date_complexe.afisare_tablouri_artist;
347

```

Script Output: Task completed in 0,041 seconds

Package PACHET_DATE_COMPLEXE compiled

Package Body PACHET_DATE_COMPLEXE compiled

PL/SQL procedure successfully completed.

ARTIST de Kooning
LISTA TABLOURI
Police Gazette
Interchange
Woman III

ARTIST Klimt
LISTA TABLOURI
Portrait of Adele Bloch-Bauer I
Portrait of Adele Bloch-Bauer II
Water Serpents II

ARTIST Pollock
LISTA TABLOURI
No. 5, 1948

ARTIST van Gogh
LISTA TABLOURI
Portrait of Dr. Gachet
Portrait of the Postman Joseph Roulin
Irises
Starry Night Over the Rhône
Self-portrait without beard

ARTIST Eakins
NU SUNT TABLOURI

ARTIST Bacon

```

332    for i in tab_seller.first..tab_seller.last
333    loop
334        select distinct b.painting_id bulk collect into my_vec_pp
335        from buying b
336        where b.seller_id = tab_seller(i);
337
338        my_tab_e.extend;
339        my_tab_e(i) := my_vec_pp;
340    end loop;
341
342 end pachet_date_complexe;
344 /
345
346 execute pachet_date_complexe.afisare_tablouri_artist;
347

```

Script Output: Task completed in 0,041 seconds

Package PACHET_DATE_COMPLEXE compiled

Package Body PACHET_DATE_COMPLEXE compiled

PL/SQL procedure successfully completed.

Woman III

ARTIST Klimt
LISTA TABLOURI
Portrait of Adele Bloch-Bauer I
Portrait of Adele Bloch-Bauer II
Water Serpents II

ARTIST Pollock
LISTA TABLOURI
No. 5, 1948

ARTIST van Gogh
LISTA TABLOURI
Portrait of Dr. Gachet
Portrait of the Postman Joseph Roulin
Irises
Starry Night Over the Rhône
Self-portrait without beard

ARTIST Eakins
NU SUNT TABLOURI

ARTIST Bacon
LISTA TABLOURI
Triptych, 1976

```
execute pachet_date_complexe.afisare_tablouri_seller;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the PL/SQL code for the package body 'PACHET_DATE_COMPLEXE'. The right pane shows the output of the executed procedure, listing various artworks and their details.

```

SQL Worksheet: History
Worksheet: Query Builder
338     my_tab_e.extend;
339     my_tab_e(i) := my_vec_pp;
340   end loop;
341
342 end pachet_date_complexe;
343 /
344
345 execute pachet_date_complexe.afisare_tablouri_artist;
346
347 execute pachet_date_complexe.afisare_tablouri_seller;
348
349 execute pachet_date_complexe.afisare_tablouri_seller_artist;
350
351 declare
352   nume artist.a_last_name%type;
353
Script Output: X | Query Result: X
Task completed in 0.064 seconds
Package Body PACHET_DATE_COMPLEXE compiled
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.

Seller Rybolovlev
LISTA TABLOURI
Salvador Mundi
Portrait of Adele Bloch-Bauer II
Rideau, Cruchon et Compotier
Portrait of the Postman Joseph Roulin

Seller Geffen
LISTA TABLOURI
Interchange
Boy with a Pipe
Portrait of Adele Bloch-Bauer I
No. 5, 1948
Dora Maar au Chat
Yo, Picasso
Police Gazette
Trises
Gioconda

Seller Embiricos
LISTA TABLOURI
The Card Players
Woman III
Le Reve

Seller Staechelin
LISTA TABLOURI

```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the PL/SQL code for the package body 'PACHET_DATE_COMPLEXE'. The right pane shows the output of the executed procedure, listing various artworks and their details.

```

SQL Worksheet: History
Worksheet: Query Builder
338     my_tab_e.extend;
339     my_tab_e(i) := my_vec_pp;
340   end loop;
341
342 end pachet_date_complexe;
343 /
344
345 execute pachet_date_complexe.afisare_tablouri_artist;
346
347 execute pachet_date_complexe.afisare_tablouri_seller;
348
349 execute pachet_date_complexe.afisare_tablouri_seller_artist;
350
351 declare
352   nume artist.a_last_name%type;
353
Script Output: X | Query Result: X
Task completed in 0.064 seconds
Package Body PACHET_DATE_COMPLEXE compiled
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.

Seller Rybolovlev
LISTA TABLOURI
Salvador Mundi
Portrait of Adele Bloch-Bauer II
Rideau, Cruchon et Compotier
Portrait of the Postman Joseph Roulin

Seller Geffen
LISTA TABLOURI
Interchange
Boy with a Pipe
Portrait of Adele Bloch-Bauer I
No. 5, 1948
Dora Maar au Chat
Yo, Picasso
Police Gazette
Trises
Gioconda

Seller Embiricos
LISTA TABLOURI
The Card Players
Woman III
Le Reve

Seller Staechelin
LISTA TABLOURI
When Will You Marry?
Women of Algiers
Young Girl with a Flower Basket
Self-portrait without beard

Seller Altman
LISTA TABLOURI
Otahai
Nude, Green Leaves and Bust
Portrait of Adele Bloch-Bauer I
Dora Maar au Chat
Les Noces de Pierrette
Water Serpents II
Portrait of Dr. Gachet
Starry Night Over the Rhône
Triptych, 1976

```

```
execute pachet_date_complexe.afisare_tablouri_seller_artist;
```

Worksheet Query Builder

```
338     my_tab_e.extend;
339     my_tab_e(i) := my_vec_pp;
340   end loop;
341
342
343 end pachet_date_complexe;
344 /
345
346 execute pachet_date_complexe.afisare_tablouri_artist;
347
348 execute pachet_date_complexe.afisare_tablouri_seller;
349
350 execute pachet_date_complexe.afisare_tablouri_seller_artist;
351
352 declare
353   nome artist.a_last_name%type;
354
355
-----
```

Seller Rybolovlev
LISTA TABLOURI
Salvador Mundi - da Vinci
Portrait of Adele Bloch-Bauer II - Klimt
Rideau, Cruchon et Compotier - Cézanne
Portrait of the Postman Joseph Roulin - van Gogh

Cel mai scump tablou vandut: Salvador Mundi

Seller Geffen
LISTA TABLOURI
Interchange - de Kooning
Boy with a Pipe - Picasso
Portrait of Adele Bloch-Bauer I - Klimt
No. 5, 1948 - Pollock
Dora Maar au Chat - Picasso
Yo, Picasso - Picasso
Police Gazette - de Kooning
Irises - van Gogh
Gioconda - da Vinci

Cel mai scump tablou vandut: Gioconda

Script Output X | Query Result X

File Edit View Insert Tools Window Help

Task completed in 0.055 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

SQL Worksheet History

Worksheet Query Builder

homeuser Buffer Size:20000

Seller Embiricos
LISTA TABLOURI
The Card Players - Gauguin
Woman III - de Kooning
Le Reve - Picasso

Cel mai scump tablou vandut: The Card Players

Seller Staechelin
LISTA TABLOURI
When Will You Marry? - Gauguin
Women of Algiers - Picasso
Young Girl with a Flower Basket - Picasso
Self-portrait without beard - van Gogh

Cel mai scump tablou vandut: When Will You Marry?

Seller Altmann
LISTA TABLOURI
Otahi - Gauguin
Nude, Green Leaves and Bust - Picasso
Portrait of Adele Bloch-Bauer I - Klimt
Dora Maar au Chat - Picasso

Script Output X Query Result X

Task completed in 0.055 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

SQL Worksheet | History

Worksheet | Query Builder

homeuser | Buffer Size: 20000

Seller Staechelin
LISTA TABLOURI
When Will You Marry? - Gauguin
Women of Algiers - Picasso
Young Girl with a Flower Basket - Picasso
Self-portrait without beard - van Gogh

Cel mai scump tablou vandut: When Will You Marry? - Gauguin

Seller Altman
LISTA TABLOURI
Otahi - Gauguin
Nude, Green Leaves and Bust - Picasso
Portrait of Adele Bloch-Bauer I - Klimt
Dora Maar au Chat - Picasso
Les Noces de Pierrette - Picasso
Water Serpents II - Klimt
Portrait of Dr. Gachet - van Gogh
Starry Night Over the Rhône - van Gogh
Triptych, 1976 - Bacon

Cel mai scump tablou vandut: Water Serpents II - Klimt

338 my_tab_e.extend;
339 my_tab_e(i) := my_vec_pp;
340 end loop;
341
342 end packet_date_complex;
343 /
345
346 execute packet_date_complex.afisare_tablouri_artist;
347
348 execute packet_date_complex.afisare_tablouri_seller;
349
350 execute packet_date_complex.afisare_tablouri_seller.artist;
351
352 declare
353 nume artist.a_last_name%type;

Script Output | Query Result | X

Task completed in 0.055 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

```
declare
    nume artist.a_last_name%type;
begin
    pachet_date_complexe.afisare_nume_artist(22, nume);
    dbms_output.put_line('Numele este ' || nume);
end;
/
```

The screenshot shows the Oracle SQL Developer interface. At the top, there's a toolbar with various icons. Below it is a tab bar with 'Worksheet' and 'Query Builder'. The main area contains a code editor with numbered lines from 351 to 361. Lines 352 through 359 are highlighted in blue, indicating they are part of a PL/SQL block. The code itself is a simple procedure that prints the last name of the 22nd artist to the screen. Below the code editor is a status bar with 'Script Output' and 'Query Result' tabs, both of which are currently inactive. The status bar also shows 'Task completed in 0,053 seconds'. At the bottom of the interface, there's a 'Dbms Output' window with a toolbar and a buffer size of 20000. The output window displays the result of the execution: 'Numele este van Gogh'.

```
declare
    nume painting.painting_title%type;
begin
    pachet_date_complexe.celMaiScump(3, nume);
    dbms_output.put_line('Tabloul este ' || nume);
end;
/
```

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Worksheet" and contains the following PL/SQL code:

```
361
362 declare
363   nume painting.painting_title%type;
364 begin
365
366   pachet_date_complexe.celMaiScump(3, nume);
367   dbms_output.put_line('Tabloul este ' || nume);
368
369 end;
370 /
371
```

Below the code, the status bar indicates "Task completed in 0,05 seconds".

PL/SQL procedure successfully completed.

The screenshot shows the Oracle SQL Developer interface with the "Dbms Output" tab selected. The output window displays the message "Tabloul este The Card Players".