

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

2.praktiskais darbs

mācību priekšmetā

“Pēcrelāciju datu bāzes sistēmas(DB2)”

**Relāciju-objektu datu bāze.**

Izstrādāja: Viktors Stepičevs

III.kurss RDBD0, 4.grupa, 191RDB100

Pārbaudīja: Prof. J. Eiduks

**2021./22/māc.gads**

**Darba saturs**

[**Darba uzdevumi** 3](#_Toc89280614)

[**Anotācija** 4](#_Toc89280615)

[**Datu bāzes struktūru shēma** 5](#_Toc89280616)

[**Tabulu un metožu definēšana** 6](#_Toc89280617)

[**Objektu tabula ar vienkāršiem objektiem** 6](#_Toc89280618)

[**Objektu tabula ar saliktiem objektiem** 7](#_Toc89280619)

[**Tabula ar objektu kolonu** 8](#_Toc89280620)

[**Tabula ar objektu kolekciju** 9](#_Toc89280621)

[**Datu ievade tabulās. Tabulu objektu sasaiste ar objektu saitēm.** 10](#_Toc89280622)

[**Objektu tabula ar vienkāršiem objektiem** 10](#_Toc89280623)

[**Objektu tabula ar saliktiem objektiem** 10](#_Toc89280624)

[**Tabula ar objektu kolonu** 11](#_Toc89280625)

[**Tabula ar objektu kolekciju** 12](#_Toc89280626)

[**Vaicājumu realizēšana** 14](#_Toc89280627)

[**Vaicājums objektu tabulai ar vienkāršiem objektiem** 14](#_Toc89280628)

[**Vaicājums objektu tabulai ar saliktiem objektiem** 16](#_Toc89280629)

[**Vaicājums tabulai ar objektu kolonām** 18](#_Toc89280630)

[**Vaicājums tabulai ar objektu kolekcijām. Vaicājums objektu tabulām ar objektu saiti 1:N** 19](#_Toc89280631)

[**Vaicājums objektu tabulām ar objektu saiti 1:1** 21](#_Toc89280632)

[**Vaicājums, izmantojot MAP tipa metodi** 22](#_Toc89280633)

[**Vaicājums, izmantojot ORDER tipa metodi** 23](#_Toc89280634)

[**Vaicājums, izmantojot MEMBER tipa metodi** 24](#_Toc89280635)

[**Metožu mantošanas demonstrēšana** 26](#_Toc89280636)

[**Secinājumi** 27](#_Toc89280637)

[**Literatūra** 28](#_Toc89280638)

[**Pielikumi** 29](#_Toc89280639)

[**Darba izmantotas datu bāzes SQL kods** 29](#_Toc89280640)

[**Darba poligona apraksts** 37](#_Toc89280641)

# **Darba uzdevumi**

1. Darbā jāizveido relāciju – objektu datu bāzes struktūras shēma, kurā ir;

1) objektu tabulas ar vienkāršiem un saliktiem objektiem;

2) tabulas ar objektu kolonām;

3) tabulas ar objektu kolekcijām;

4) objektu saites 1 ar 1 un 1 ar N (daudziem);

5) objektiem ir MAP, ORDER un MEMBER tipa metodes.

2. Darbā obligāti jāiekļauj datu bāzes struktūras shēma (tabulas, objekti, atribūti, saites, metodes).

3. Jādefinē tabulas un metodes.

4. Jāveic datu ievade tabulās.

5. Jāveic tabulu objektu sasaiste ar objektu saitēm.  
6. Vaicājumu realizēšana relāciju-objektu datu bāzei:

1) vaicājumi objektu tabulai ar vienkāršiem un saliktiem objektiem;

2) vaicājumi tabulai ar kolonas tipa objektiem;  
3) vaicājumi tabulai ar objektu kolekciju, TABLE() funkcija;  
4) vaicājums objektu tabulām ar objektu saiti viens ar vienu 1 : 1.  
5) vaicājumi objektu tabulām ar saiti viens ar daudziem 1 : N;  
6) vaicājumi, izmantojot objektu metodes.

7. Metožu mantošanas demonstrēšana (neobligāti).

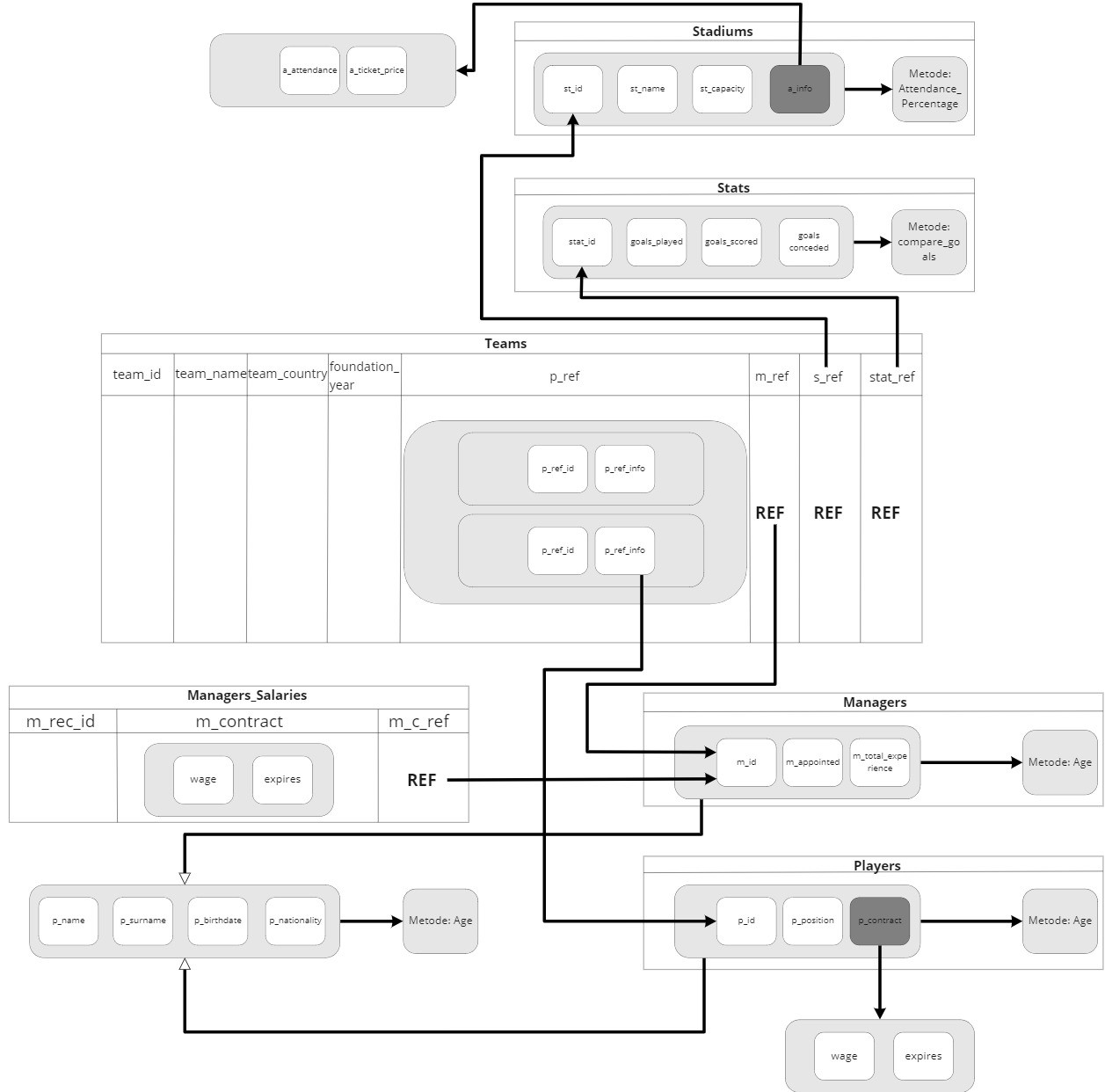
8. Objektu skati ar metodēm (neobligāti).  
9. Secinājumi par darbu (ne kas izdarīts, bet kas noskaidrots, kādas ir apskatītās tehnoloģijas labās un sliktās īpašības).

# **Anotācija**

Darbā tiks apskatītas relāciju-objektu datu bāzes pamatjēdzieni sporta komandu datu bāzēs sistēmās piemērā, definējot objektu tabulas, tabulas ar objektu kolonu un kolekcijām, objektu metodes un metožu mantošanu, kā arī tika veikta objektu sasaiste. Datu bāze pilnībā tika izstrādāta Oracle SQL Developer vidē.

# **Datu bāzes struktūru shēma**

Pirmkārt, izpētot darba uzdevumus un sākot datu bāzes projektēšanu, par datu bāzes problemsfēru es pieņēmu turpināt ar sporta tematiku. Lai realizēta datu bāze atbilstu visiem nepieciešamiem kritējiem, es nedaudz pārdefinēju datu bāzes ideju. Definēta datu bāze satur informāciju par dažam komandām, precīzāk sākot to spēlētājiem, treneriem, un šo sezona statistiku.



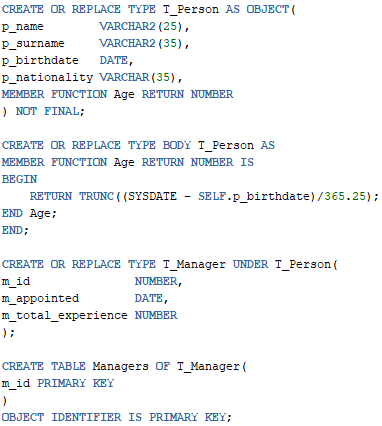
Kā ir izveidotājā shēmā, darbā tika realizētās 6 pamata tabulas, divas objektu tabulas ar vienkāršiem objektiem Managers un Stats, objektu tabulas ar saliktiem objektiem Players un Stadiums, tabula ar objektu kolonu Manager\_Salaries un tabula ar objektu kolekciju Teams, kur objektu kolekcija ir atsauces tabula Players\_ref.

# **Tabulu un metožu definēšana**

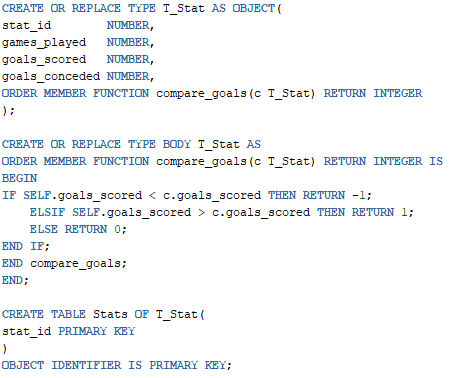
Pēc relāciju-objekta datu bāzes shēmas izveides tika realizēti visi objektu, tabulas, metodes un saites. Darba gaitā izveidotas datu bāzes kods pilnībā ir pieejams darba pielikumos.

## **Objektu tabula ar vienkāršiem objektiem**

Datu bāzē tika definēta divas objektu tabula ar vienkāršo objektu – treneru tabula Managers un komandu statistikas tabula Stats.



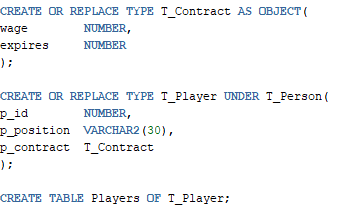
Kā ir redzams, tabula ir izveidota objekta T\_Manager pamatā. Šeit tika pielietota mantošanas konstrukcija, kur objekts T\_Manager manto objekta T\_Person atribūtus un metodes. Objekts T\_Person glāb iekša cilvēka vārdu, uzvārdu, dzimšanas datumu un nacionalitāti, kā arī MEMBER tipa metodi Age, kurš noteic katra cilvēka vecumu. Savukārt, T\_Manager objekts pārmanto T\_Person objekta atribūtus un Age metodi, kā arī satur trenera identifikatoru, datumu, kad treneris tika izvēlēts komandā savā amatā un trenera kopējo pieredzes gadu skaits. Izveidojot objekta tabulu, trenera identifikators tiek definēts par objekta identifikatoru ar “OBJECT IDENTIFIER IS PRIMARY KEY” palīdzību, lai turpmāk citā tabulā būtu iespējams izvedot referenci uz šo tabulu.



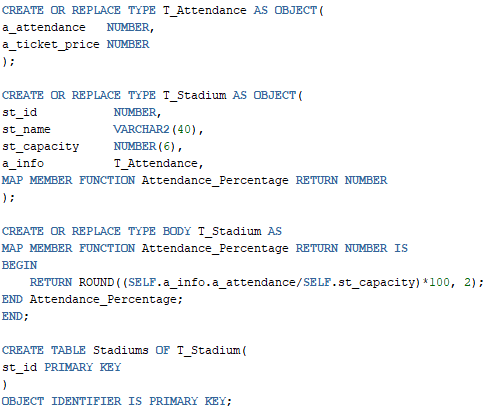
Savukārt, tabula Stats tika izveidota objekta T\_Stat pamatā, kurš satur objekta identifikatoru, glābj informāciju par komandas aizvadīto spēļu skaitu, iegūto un ielaisto vārtu skaitu, kā arī iekša tika definēta ORDER tipa metode compare\_goals, ar kuru palīdzību ir iespējams salīdzināt, kuram no komandām relatīvi vienai izvelētai ir vairāk vai mazāk iegūto vārtu.

## **Objektu tabula ar saliktiem objektiem**

Lai pilnvērtīgi realizētu visus nepieciešamus darba posmus, darbā es definēju divas objektu tabulas ar saliktiem objektiem – komandu spēlētāju tabula Players ar komandu stadionu tabulu Stadiums.



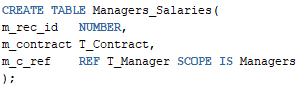
Objektu tabula Players tika izveidota objekta T\_Player bāzē. Līdzīgi T\_Manager objektam, T\_Player pārmanto T\_Person atribūtus un metodi Age, kā arī satur spēlētāja identifikatoru, spēlētāja pozīciju un objektu T\_Contract, kas satur informāciju par kontraktu - atalgojumu un gadu, līdz kuram kontrakts ir derīgs.



Savukārt, līdzīgi tabulai Managers, tabula Stadiums tiek definēta ar objekta identifikātoru, lai vēlāk izveidot referenci. Tā tiek izveidota uz objekta T\_Stadium pamata, kas satur stadiona identifikatoru, stadiona nosaukumu, kapacitāti un objektu T\_Attendance, kas satur stadiona apmeklēšanas statistiku un biļešu cenas. Papildus, objektam T\_Stadium ir definēta MAP tipa metode Attendance\_Percentage, kas aprēķina stadiona aizpildījumu.

## **Tabula ar objektu kolonu**

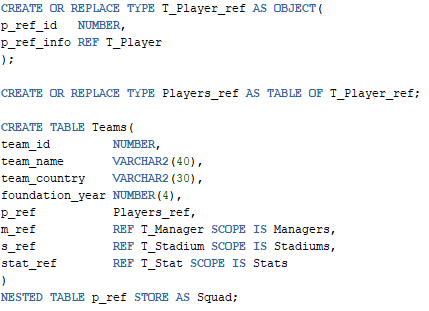
Tabulai ar treneru atalgojumu es pienācu nedaudz savādāk, nekā Players tabulā. Šajā gadījumā es izveidoju relāciju tabulu, kur objekts T\_Contract glābās ka tabulas kolona.



Tabula satur ieraksta identifikatoru, kā arī pati objektu kolonu un atsauci(REF) uz objektu T\_Manager, lai referencētu treneri.

## **Tabula ar objektu kolekciju**

Datu bāzē tika realizēta viena tabula ar objektu kolekciju, komandu tabula Teams.



Tabulas atribūti ir komandas identifikators, nosaukums, valsts, kurā komanda spēlē, dibinašanas gads, objektu kolekcija Players\_ref, kura kopumā ir tabulas struktūra, kura glāba atsauces uz tabulu Players (katrai komandai ir daži spēlētāji) un atsauces uz tabulām Managers, Stadiums un Stats, lai tieši referencētu katras komandas datus.

# **Datu ievade tabulās. Tabulu objektu sasaiste ar objektu saitēm.**

Šajā sadaļa es pieņēmu apvienot divus darba uzdevumus, jo būtībā objektu sasaistes es atrādīju augstāk, bet pie datu ievades to visu varētu skaidri redzēt un būtu vieglāk paskaidrot. Tabulās datus es ievadīju vienkārši izmantojot INSERT INTO komandu. Lai viss būtu labāk saredzams, datu ievadi es ievietošu ka tekstu, nevis bildi.

## **Objektu tabula ar vienkāršiem objektiem**

**INSERT INTO Managers VALUES(T\_Manager('Jurgen', 'Klopp', TO\_DATE('16-06-1967', 'DD-MM-YYYY'), 'German', '1', TO\_DATE('08-10-2015', 'DD-MM-YYYY'), '20'));**

**INSERT INTO Managers VALUES(T\_Manager('Xavi', 'Hernandez', TO\_DATE('25-01-1980', 'DD-MM-YYYY'), 'Spanish', '2', TO\_DATE('06-11-2021', 'DD-MM-YYYY'), '2'));**

**INSERT INTO Managers VALUES(T\_Manager('Carlo', 'Anchelotti', TO\_DATE('10-06-1959', 'DD-MM-YYYY'), 'Italian', '3', TO\_DATE('01-06-2021', 'DD-MM-YYYY'), '26'));**

**INSERT INTO Managers VALUES(T\_Manager('Ole Gunnar', 'Solskjaer', TO\_DATE('26-02-1973', 'DD-MM-YYYY'), 'Norwegian', '4', TO\_DATE('19-12-2018', 'DD-MM-YYYY'), '10'));**

**INSERT INTO Managers VALUES(T\_Manager('Massimiliano', 'Allegri', TO\_DATE('11-08-1967', 'DD-MM-YYYY'), 'Italian', '5', TO\_DATE('28-05-2021', 'DD-MM-YYYY'), '18'));**

**INSERT INTO Managers VALUES(T\_Manager('Stefano', 'Pioli', TO\_DATE('20-10-1965', 'DD-MM-YYYY'), 'Italian', '6', TO\_DATE('09-10-2019', 'DD-MM-YYYY'), '18'));**

**INSERT INTO Stats VALUES(T\_Stat('1', '13', '21', '22'));**

**INSERT INTO Stats VALUES(T\_Stat('2', '13', '39', '11'));**

**INSERT INTO Stats VALUES(T\_Stat('3', '14', '30', '18'));**

**INSERT INTO Stats VALUES(T\_Stat('4', '14', '18', '16'));**

**INSERT INTO Stats VALUES(T\_Stat('5', '14', '34', '15'));**

**INSERT INTO Stats VALUES(T\_Stat('6', '14', '23', '16'));**

Būtībā te nav daudz ko varētu pastāstīt. Ir redzams, ka pie ievades tabulā figurē tikai viens objekts.

## **Objektu tabula ar saliktiem objektiem**

**INSERT INTO Players VALUES(T\_Player('Cristiano', 'Ronaldo', TO\_DATE('05-02-1985', 'DD-MM-YYYY'), 'Portugese', '1', 'Forward', T\_Contract('430000', '2023')));**

**INSERT INTO Players VALUES(T\_Player('Paul', 'Pogba', TO\_DATE('15-03-1993', 'DD-MM-YYYY'), 'French', '2', 'Midfielder', T\_Contract('340000', '2022')));**

**INSERT INTO Players VALUES(T\_Player('Bruno', 'Fernandes', TO\_DATE('08-09-1994', 'DD-MM-YYYY'), 'Portugese', '3', 'Midfielder', T\_Contract('210000', '2025')));**

**INSERT INTO Players VALUES(T\_Player('Virgil', 'van Dijk', TO\_DATE('08-07-1991', 'DD-MM-YYYY'), 'Dutch', '4', 'Defender', T\_Contract('260000', '2025')));**

**INSERT INTO Players VALUES(T\_Player('Mohamed', 'Salah', TO\_DATE('15-06-1992', 'DD-MM-YYYY'), 'Egyptian', '5', 'Forward', T\_Contract('235000', '2023')));**

**INSERT INTO Players VALUES(T\_Player('Allison', 'Becker', TO\_DATE('02-10-1992', 'DD-MM-YYYY'), 'Brazilian', '6', 'Goalkeeper', T\_Contract('175000', '2027')));**

**INSERT INTO Players VALUES(T\_Player('Zlatan', 'Ibrahimovic', TO\_DATE('03-10-1981', 'DD-MM-YYYY'), 'Swedish', '7', 'Forward', T\_Contract('205000', '2022')));**

**INSERT INTO Players VALUES(T\_Player('Olivier', 'Giroud', TO\_DATE('30-09-1986', 'DD-MM-YYYY'), 'French', '8','Forward', T\_Contract('100000', '2023')));**

**INSERT INTO Players VALUES(T\_Player('Fikayo', 'Tomori', TO\_DATE('19-12-1997', 'DD-MM-YYYY'), 'English', '9', 'Defender', T\_Contract('60000', '2025')));**

**INSERT INTO Players VALUES(T\_Player('Wojciech', 'Szczesny', TO\_DATE('18-04-1990', 'DD-MM-YYYY'), 'Polish', '10', 'Goalkeeper', T\_Contract('232000', '2024')));**

**INSERT INTO Players VALUES(T\_Player('Leonardo', 'Bonucci', TO\_DATE('01-05-1987', 'DD-MM-YYYY'), 'Italian', '11', 'Defender', T\_Contract('211000', '2024')));**

**INSERT INTO Players VALUES(T\_Player('Manuel', 'Locatelli', TO\_DATE('08-01-1998', 'DD-MM-YYYY'), 'Italian', '12', 'Midfielder', T\_Contract('117000', '2026')));**

**INSERT INTO Players VALUES(T\_Player('Karim', 'Benzema', TO\_DATE('19-08-1987', 'DD-MM-YYYY'), 'French', '13', 'Forward', T\_Contract('320000', '2023')));**

**INSERT INTO Players VALUES(T\_Player('Vinicius', 'Jr', TO\_DATE('12-07-2000', 'DD-MM-YYYY'), 'Brazilian', '14', 'Forward', T\_Contract('215000', '2025')));**

**INSERT INTO Players VALUES(T\_Player('Luka', 'Modric', TO\_DATE('09-09-1985', 'DD-MM-YYYY'), 'Croatian', '15', 'Midfielder', T\_Contract('322000', '2022')));**

**INSERT INTO Players VALUES(T\_Player('Gerard', 'Pique', TO\_DATE('02-02-1987', 'DD-MM-YYYY'), 'Spanish', '16', 'Defender', T\_Contract('120000', '2024')));**

**INSERT INTO Players VALUES(T\_Player('Marc-Andre', 'ter Stegen', TO\_DATE('30-04-1992', 'DD-MM-YYYY'), 'German', '17', 'Goalkeeper', T\_Contract('165000', '2025')));**

**INSERT INTO Players VALUES(T\_Player('Frenkie', 'de Jong', TO\_DATE('12-05-1997', 'DD-MM-YYYY'), 'Dutch', '18', 'Midfielder', T\_Contract('415000', '2024')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('1', 'San Siro', '80018', T\_Attendance('59018', '75')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('2', 'Estadio Santiago Bernabeu', '81044', T\_Attendance('62890', '130')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('3', 'Old Trafford', '76000', T\_Attendance('75800', '120')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('4', 'Allianz Stadium', '41507', T\_Attendance('38243', '85')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('5', 'Camp Nou', '99354', T\_Attendance('69071', '110')));**

**INSERT INTO Stadiums VALUES(T\_Stadium('6', 'Anfield', '53394', T\_Attendance('53000', '115')));**

Pie ievades šīs divās tabulās varētu novērot, ka iekš objekta, kurā pamatā bija definētas objektu tabulas ir vel viens objekts, kas un padara šīs tabulas par tabulām ar saliktiem objektiem.

## **Tabula ar objektu kolonu**

**INSERT INTO Managers\_Salaries VALUES('1', T\_Contract('230000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 1));**

**INSERT INTO Managers\_Salaries VALUES('2', T\_Contract('150000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 2));**

**INSERT INTO Managers\_Salaries VALUES('3', T\_Contract('200000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 3));**

**INSERT INTO Managers\_Salaries VALUES('4', T\_Contract('205000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 4));**

**INSERT INTO Managers\_Salaries VALUES('5', T\_Contract('140000', '2025'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 5));**

**INSERT INTO Managers\_Salaries VALUES('6', T\_Contract('135000', '2022'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 6));**

Šajā tabulā ir iespējams novērot, ka par vienu kolonu kalpo objekts T\_Contract, kā arī tiek definēta reference uz tabulu Managers, izpildot SELECT vaicājumu un pēc identifikatora atrādot nepieciešamo ierakstu.

## **Tabula ar objektu kolekciju**

**DECLARE**

**TYPE T\_Player\_arr IS VARRAY(18) OF REF T\_Player;**

**Players\_arr T\_Player\_arr := T\_Player\_arr();**

**BEGIN**

**Players\_arr.EXTEND(18);**

**FOR i IN 1..18 LOOP**

**SELECT REF(p) INTO Players\_arr(i) FROM Players p WHERE p.p\_id = (i);**

**END LOOP;**

**INSERT INTO Teams VALUES('1', 'Manchester United', 'England', '1878', Players\_ref(T\_Player\_ref('1', Players\_arr(1)),**

**T\_Player\_ref('2', Players\_arr(2)),**

**T\_Player\_ref('3', Players\_arr(3))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 4),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 3),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 1));**

**INSERT INTO Teams VALUES('2', 'Liverpool FC', 'England', '1892', Players\_ref(T\_Player\_ref('4', Players\_arr(4)),**

**T\_Player\_ref('5', Players\_arr(5)),**

**T\_Player\_ref('6', Players\_arr(6))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 1),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 6),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 2));**

**INSERT INTO Teams VALUES('3', 'AC Milan', 'Italy', '1899', Players\_ref(T\_Player\_ref('7', Players\_arr(7)),**

**T\_Player\_ref('8', Players\_arr(8)),**

**T\_Player\_ref('9', Players\_arr(9))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 6),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 1),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 3));**

**INSERT INTO Teams VALUES('4', 'Juventus', 'Italy', '1897', Players\_ref(T\_Player\_ref('10', Players\_arr(10)),**

**T\_Player\_ref('11', Players\_arr(11)),**

**T\_Player\_ref('12', Players\_arr(12))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 5),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 4),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 4));**

**INSERT INTO Teams VALUES('5', 'Real Madrid', 'Spain', '1902', Players\_ref(T\_Player\_ref('13', Players\_arr(13)),**

**T\_Player\_ref('14', Players\_arr(14)),**

**T\_Player\_ref('15', Players\_arr(15))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 3),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 2),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 5));**

**INSERT INTO Teams VALUES('6', 'FC Barcelona', 'Spain', '1899', Players\_ref(T\_Player\_ref('16', Players\_arr(16)),**

**T\_Player\_ref('17', Players\_arr(17)),**

**T\_Player\_ref('18', Players\_arr(18))),**

**(SELECT REF(m) FROM Managers m WHERE m.m\_id = 2),**

**(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 5),**

**(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 6));**

**END;**

Šajā gadījumā datus bija jāievada ar PL/SQL procedūras palīdzību. Lai nedaudz automatizētu procesu un nedefinētu atsauces uz katru spēlētāju, jo gadījumos kad Jums vajag izmantot vairākas atsauce tas vienkārši aizņems vairāk laiku un rādās lielāka iespēja kļūdām, es definēju masīvu, kur katrā elementā glabās spēlētāja atsauce. Ar LOOP palīdzību es ievietoju tās atsauces pozīcijās līdzīgām spēlētāju identifikatoram tabulas struktūrā Players\_ref un vēlāk, ievietojot datus Teams tabulā, rindā, kur jāatrodas objektu kolekcijai, ir jāievieto atbilstošus masīva elementus. Arī ir redzams, ka līdzīgi Managers\_Salaries tabulai šeit ir vairākas atsauces uz tabulām Managers, Stadiums un Stats, kurās strādā tā pati, kā aprakstīju to iepriekš.

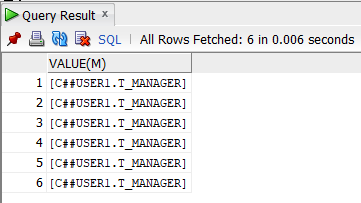
# **Vaicājumu realizēšana**

## **Vaicājums objektu tabulai ar vienkāršiem objektiem**

Kopumā, varētu secināt ka objektu tabula ir tāda tabula, kur katra no to rindām ir kāds objekts. To arī varētu novērot izmantojot vaicājumu ar funkciju VALUE. Izvadot visu informāciju no objektu tabulas, būs redzams objekta tipa nosaukumi, ko var novērot ar sekojošiem vaicājumiem.

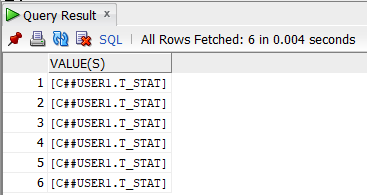
**SELECT VALUE(m)**

**FROM Managers m;**



**SELECT VALUE(s)**

**FROM Stats s;**

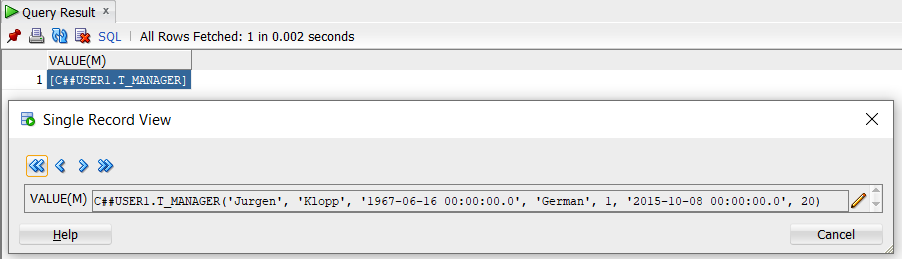


Izgūt datus no objekta tabulas ir iespējams ne tikai, ka ierasts, bet arī apskatot Single Record View, kas atrāda ieraksta faktiskas vērtības.

**SELECT VALUE(m)**

**FROM Managers m**

**WHERE VALUE(m).m\_id = 1;**



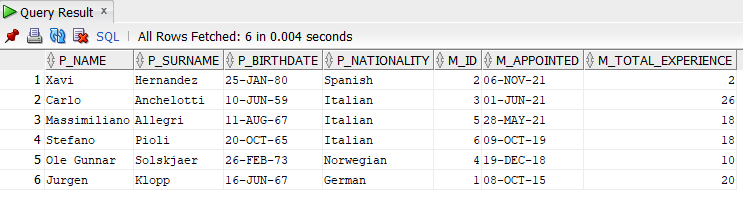
Nākamie vaicājumi atrādīs, ka ierasta veidā tiek izgūta informācija no objektu tabulām.

Šeit tika izvadītā visa informācija no tabulas Managers un dati tika atlasīti pēc treneru iestāšanos amatā.

**SELECT m.\***

**FROM Managers m**

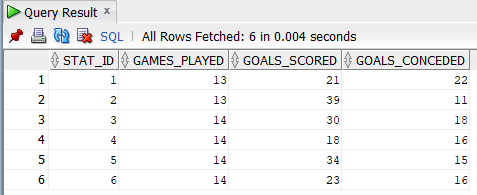
**ORDER BY m\_appointed DESC;**



Visi dati no tabulas Stats

**SELECT s.\***

**FROM Stats s;**



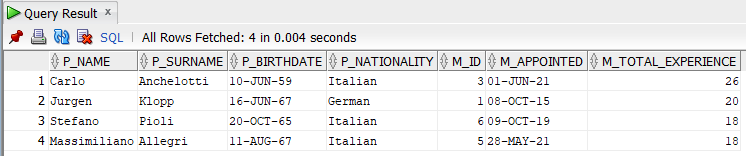
Treneri ar pieredzi, kas ir vairākā par 15 gadiem, izvadot datus dilstošā secībā.

**SELECT m.\***

**FROM Managers m**

**WHERE m.m\_total\_experience > 15**

**ORDER BY m\_total\_experience DESC;**

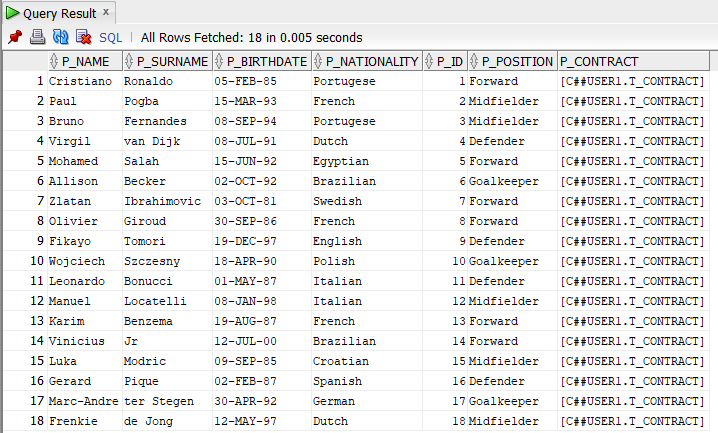


## **Vaicājums objektu tabulai ar saliktiem objektiem**

Būtība, vienīga atšķirība veicot vaicājumus objektu tabulām ar saliktiem objektiem ir tā, ka izvadot datus viena no kolonam būs iekšējais objekts.

**SELECT p.\***

**FROM Players p;**



Vel viena atšķirība objekta tabulām no relāciju tabulām ir tā, ka tām tiek ģenerēti objekta identifikatori OID.

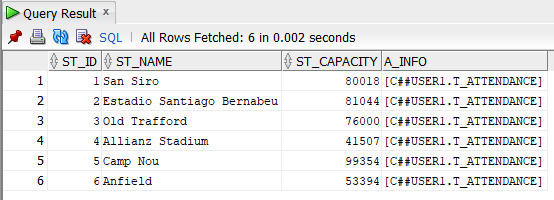
**SELECT OBJECT\_ID, p.\***

**FROM Players p;**



**SELECT s.\***

**FROM Stadiums s;**



Lai izvadītu informāciju no pakārtota objekta ir jāizmanto VALUE funkciju kombinācija ar norādi uz tabulas atribūtu, kas glāb iekša objektu.

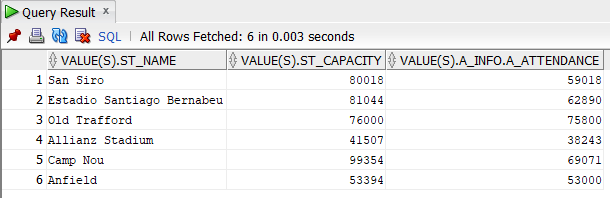
**SELECT VALUE(p).p\_name, VALUE(p).p\_surname, VALUE(p).p\_contract.wage, VALUE(p).p\_contract.expires**

**FROM Players p;**



**SELECT VALUE(s).st\_name, VALUE(s).st\_capacity, VALUE(s).a\_info.a\_attendance**

**FROM Stadiums s;**

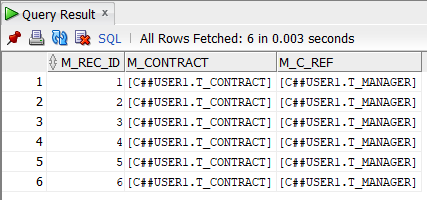


## **Vaicājums tabulai ar objektu kolonām**

Būtība, tabulā ar objektu kolonu, lai ierastā veidā izvadītu datus no tabulas, pieeja ir līdzīga kā bija aprakstīts augstāk, vienīgais relāciju tabulā to nav iespējas izdarīt ar VALUE funkcijas palīdzību, SELECT vaicājuma ir jāizmanto norādi uz tabulu.

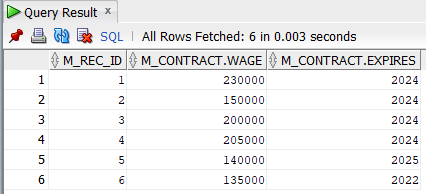
**SELECT m.\***

**FROM Managers\_Salaries m;**



**SELECT m.m\_rec\_id, m.m\_contract.wage, m.m\_contract.expires**

**FROM Managers\_ Salaries m;**

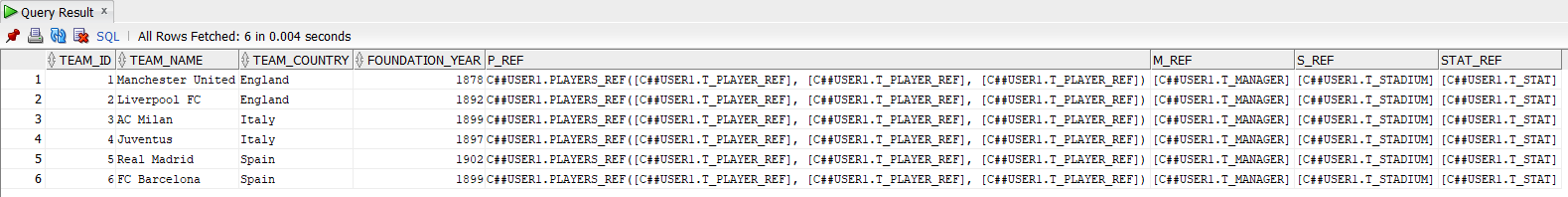


## **Vaicājums tabulai ar objektu kolekcijām. Vaicājums objektu tabulām ar objektu saiti 1:N**

Izvadot datus no tabulas Teams ir iespējams novērtot vairākas norādes uz objektiem, ko precīzāk jau aprakstīju. Savukārt, arī varētu būt redzama objektu kolekcija Players\_ref, kas norāda uz tabulu Players, attiecinot spēlētājus komandās.

**SELECT t.\***

**FROM Teams t;**



Lai izgūtu datus no objektu kolekcijas ir jāizmanto funkciju TABLE, pārveidojot to par kolekcijas objekta skatu.

**SELECT t.team\_name, VALUE(p)**

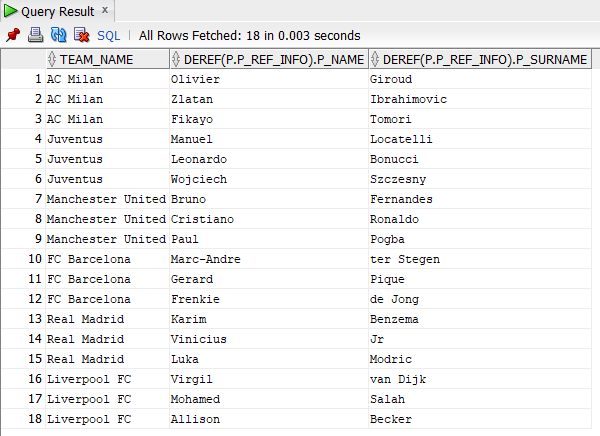
**FROM Teams t, TABLE(t.p\_ref) p;**



Ar funkcijas DEREF palīdzību mums rādās iespēja atgriezt argumenta objekta atsauci, lai turpmāk izvadītu datus no kolekcijas objekta skata.

**SELECT t.team\_name, DEREF(p.p\_ref\_info).p\_name, DEREF(p.p\_ref\_info).p\_surname**

**FROM Teams t, TABLE(t.p\_ref) p;**



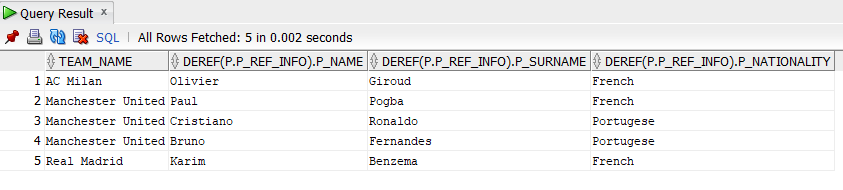
Ar šo vaicājumu tiek parbaudīta iespēja atlasīt izvadītus datus pēc kolekcijas objekta skata izvadītājiem datiem. Tiek izvadītie tie spēlētāji, kuri ir no Francijas vai Portugāle, izvadot tos pēc komandām alfabētiskajā secībā.

**SELECT t.team\_name, DEREF(p.p\_ref\_info).p\_name, DEREF(p.p\_ref\_info).p\_surname, DEREF(p.p\_ref\_info).p\_nationality**

**FROM Teams t, TABLE(t.p\_ref) p**

**WHERE DEREF(p.p\_ref\_info).p\_nationality IN ('French', 'Portugese')**

**ORDER BY team\_name;**

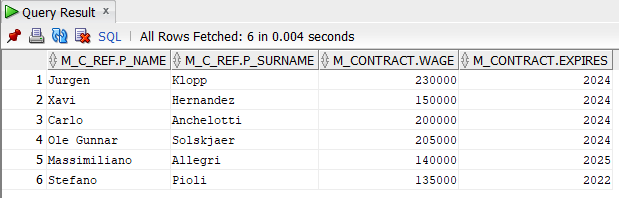


## **Vaicājums objektu tabulām ar objektu saiti 1:1**

Kopumā, lai iegūtu datus no tabulām, uz kuram tika veikta atsauce, vaicājumā ir jānorāda atribūtu, kurš referencē nepieciešamo objekta tabulu.

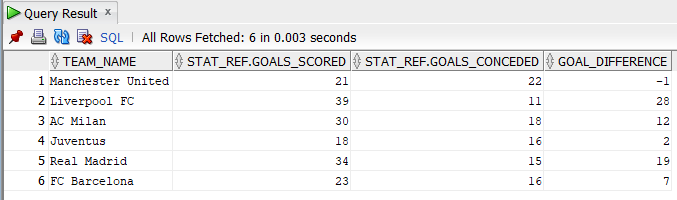
**SELECT m.m\_c\_ref.p\_name, m.m\_c\_ref.p\_surname, m.m\_contract.wage, m.m\_contract.expires**

**FROM Managers\_Salary m;**



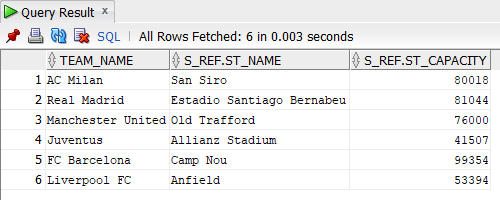
**SELECT t.team\_name, t.stat\_ref.goals\_scored, t.stat\_ref.goals\_conceded, (t.stat\_ref.goals\_scored - t.stat\_ref.goals\_conceded) AS goal\_difference**

**FROM Teams t;**



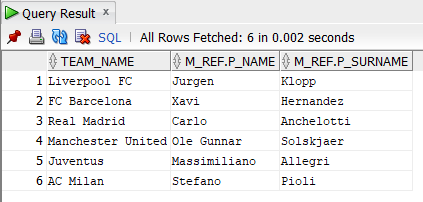
**SELECT t.team\_name, t.s\_ref.st\_name, t.s\_ref.st\_capacity**

**FROM Teams t;**



**SELECT t.team\_name, t.m\_ref.p\_name, t.m\_ref.p\_surname**

**FROM Teams t;**

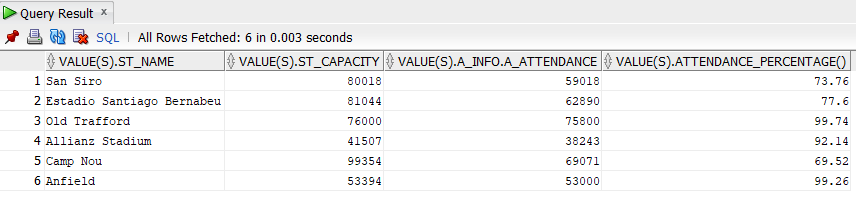


## **Vaicājums, izmantojot MAP tipa metodi**

Sekojošie vaicājumi atrāda MAP metodes Attendance\_Percentage() metodes darbību. Arī vēlējos pārbaudīt, vai metode strādās piekļūvot tai no tabulas Teams ar atsauci uz Stadiums tabulu, kā arī pārbaudīt, kā strādā filtrēšana ar metodes palīdzību, atradot stadionus ar apmeklējamību 80% un vairāk.

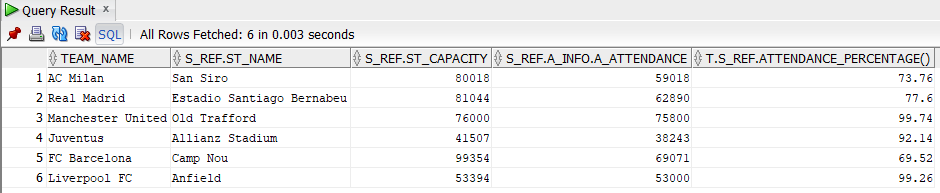
**SELECT VALUE(s).st\_name, VALUE(s).st\_capacity, VALUE(s).a\_info.a\_attendance, VALUE(s).Attendance\_Percentage()**

**FROM Stadiums s;**



**SELECT t.team\_name, t.s\_ref.st\_name, t.s\_ref.st\_capacity, t.s\_ref.a\_info.a\_attendance, t.s\_ref.Attendance\_Percentage()**

**FROM Teams t;**

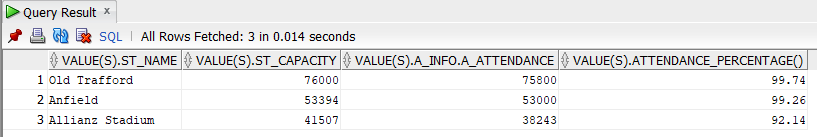


**SELECT VALUE(s).st\_name, VALUE(s).st\_capacity, VALUE(s).a\_info.a\_attendance, VALUE(s).Attendance\_Percentage()**

**FROM Stadiums s**

**WHERE VALUE(s).Attendance\_Percentage() > 80**

**ORDER BY VALUE(s).Attendance\_Percentage() DESC;**



## **Vaicājums, izmantojot ORDER tipa metodi**

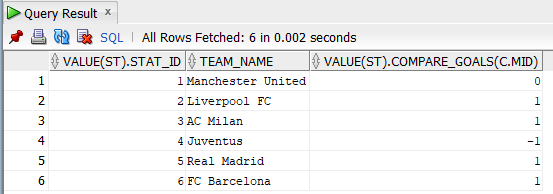
Ar ORDER TIPA metodes vaicājumu, es vēlojos noteikt, kurām no komandām ir vairāk un mazāk iegūto vārtu nekā komandai ar komandas identifikatoru 1. Gadījumā, ja iegūta vērtība ir 1, tas nozīme, ka tai komandai ir vairāk vārtu, bet ar vērtību -1 mazāk (0 vērtība ir komandai, kurai ir līdzīgs vārtu skaits).

**SELECT VALUE(st).stat\_id, t.team\_name, VALUE(st).compare\_goals(c.mid)**

**FROM Stats st, Teams t, (SELECT VALUE(stat) AS mid**

**FROM Stats stat WHERE VALUE(stat).stat\_id = '1')c**

**WHERE team\_id = stat\_id;**

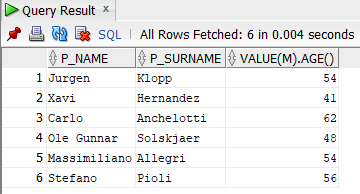


## **Vaicājums, izmantojot MEMBER tipa metodi**

Ar MEMBER tipa metodi Age(), kas noteic cilvēka vecumu no dzimšanas datuma, ir iespējams pielietot treneriem un spēlētājiem, jo šī metode tiek pārmantota no objekta T\_Person. Arī pārbaudīju metodes darbību, izsaucot to no Teams tabulas ar atsauce palīdzību.

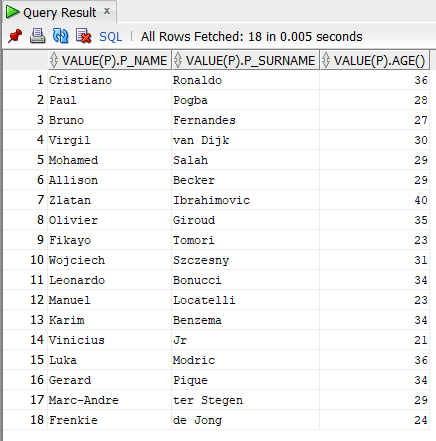
**SELECT m.p\_name, m.p\_surname, VALUE(m).Age()**

**FROM Managers m;**



**SELECT VALUE(p).p\_name, VALUE(p).p\_surname, VALUE(p).Age()**

**FROM Players p;**

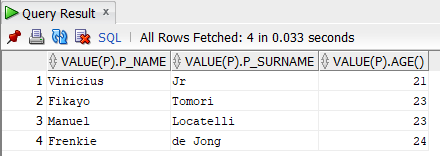


**SELECT VALUE(p).p\_name, VALUE(p).p\_surname, VALUE(p).Age()**

**FROM Players p**

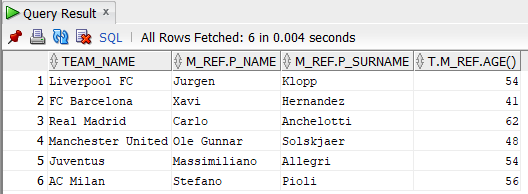
**WHERE VALUE(p).Age() < 25**

**ORDER BY VALUE(p).Age() ASC;**



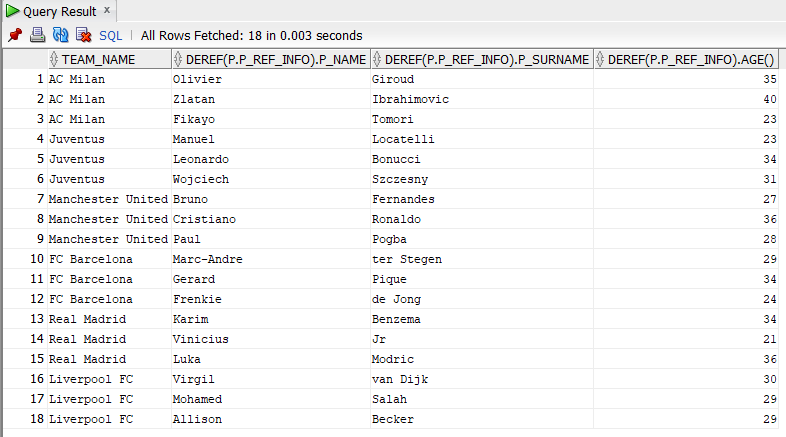
**SELECT t.team\_name, t.m\_ref.p\_name, t.m\_ref.p\_surname, t.m\_ref.Age()**

**FROM Teams t;**



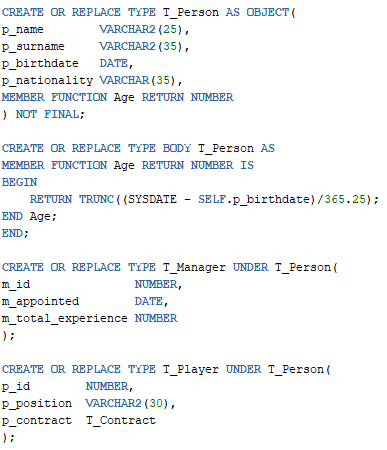
**SELECT t.team\_name, DEREF(p.p\_ref\_info).p\_name, DEREF(p.p\_ref\_info).p\_surname, DEREF(p.p\_ref\_info).Age()**

**FROM Teams t, TABLE(t.p\_ref) p;**



# **Metožu mantošanas demonstrēšana**

Būtība visu jau pastāstīju augšā, bet kopumā objekti T\_Manager un T\_Player manto metodes T\_Person atribūtus un metodi Age(). Visus vaicājumus metožu mantošanas pārbaudei es jau izpildīju iepriekš.



# **Secinājumi**

Analizējot pavadīto darba laiku un rezultātu, es varu secināt, kā kopuma viss nepieciešamais man veiksmīgi sanāca. Darbā gaitā es pirmo reizi praktiski pielietoju tādas datu bāzes tehnoloģijas ka objekti, objektu tabulas, objektu metodes un metožu mantošana.

Kopumā, runājot par grūtībām, kuras paradījās man darba izpildes gaitā, vairāk tās bija saistītas ar datu bāzes struktūru, precīzāk sākot kā datu bāzi jāprojektē, lai ieintegrētu visas nepieciešamas tehnoloģijas. Bet pamazam uzsākot darbu, visu bija definēt vieglāk un savukārt arī pētīšanu un darba izpildi bija vieglāk virzīt uz priekšu. Runājot par jaunām apskatāmām tehnoloģijām, būtībā daudz problēmas man tas nesagādāja, jo pats princips jau ir saprotams no iepriekšējas pieredzes, kas atviegloja darba uzdevumu saprašanu un darba procesa virzīšanu uz priekšu. Ja runājot par tehnoloģiju pielietojamību, manuprāt, objekti un mantošana ļoti atvieglo vairākus datu glabāšanas procesus datu bāze, jo bieži gadās situācijas, kad kādās no tabulām ir līdzīgie atribūti, un pielietojot objektus un mantošanu arī darba kods kļūst pārprotamāks.

Beigu beigās, rezultātā sanāca, ka ar darbu esmu aizkavējies, par ko es atvainojos, jo bija dažas problēmas, kuru dēļ es nespēju darbu līdz galam pabeigt un nodot laikā, bet kopumā, man patika strādāt pildot šo praktisko darbu un esmu pārliecināt, ka iegūtās zināšanas man būs noderīgas nākošo darbu izpildē, ka arī vispār nākotnē.

# **Literatūra**

1. Prof. J. Eiduks. Lekciju konspekti priekšmetā “Pēcrelāciju datu bāzes sistēmas (DB2)”
2. <https://datubaze.wordpress.com/>
3. <https://docs.oracle.com/database/121/ADOBJ/toc.htm>

# **Pielikumi**

## **Darba izmantotas datu bāzes SQL kods**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Objektu tabula Stadiums ar saliktu objektu T\_Stadium(T\_Attendance)

MAP metode Attendance\_Percentage()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Attendance AS OBJECT(

a\_attendance NUMBER,

a\_ticket\_price NUMBER

);

CREATE OR REPLACE TYPE T\_Stadium AS OBJECT(

st\_id NUMBER,

st\_name VARCHAR2(40),

st\_capacity NUMBER(6),

a\_info T\_Attendance,

MAP MEMBER FUNCTION Attendance\_Percentage RETURN NUMBER

);

CREATE OR REPLACE TYPE BODY T\_Stadium AS

MAP MEMBER FUNCTION Attendance\_Percentage RETURN NUMBER IS

BEGIN

RETURN ROUND((SELF.a\_info.a\_attendance/SELF.st\_capacity)\*100, 2);

END Attendance\_Percentage;

END;

CREATE TABLE Stadiums OF T\_Stadium(

st\_id PRIMARY KEY

)

OBJECT IDENTIFIER IS PRIMARY KEY;

INSERT INTO Stadiums VALUES(T\_Stadium('1', 'San Siro', '80018', T\_Attendance('59018', '75')));

INSERT INTO Stadiums VALUES(T\_Stadium('2', 'Estadio Santiago Bernabeu', '81044', T\_Attendance('62890', '130')));

INSERT INTO Stadiums VALUES(T\_Stadium('3', 'Old Trafford', '76000', T\_Attendance('75800', '120')));

INSERT INTO Stadiums VALUES(T\_Stadium('4', 'Allianz Stadium', '41507', T\_Attendance('38243', '85')));

INSERT INTO Stadiums VALUES(T\_Stadium('5', 'Camp Nou', '99354', T\_Attendance('69071', '110')));

INSERT INTO Stadiums VALUES(T\_Stadium('6', 'Anfield', '53394', T\_Attendance('53000', '115')));

SELECT VALUE(s)

FROM Stadiums s;

SELECT s.\*

FROM Stadiums s;

SELECT VALUE(s).st\_name, VALUE(s).st\_capacity, VALUE(s).a\_info.a\_attendance

FROM Stadiums s;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Objektu tabula Managers ar vienkarsu objektu T\_Person(Age()) <- T\_Manager

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Person AS OBJECT(

p\_name VARCHAR2(25),

p\_surname VARCHAR2(35),

p\_birthdate DATE,

p\_nationality VARCHAR(35),

MEMBER FUNCTION Age RETURN NUMBER

) NOT FINAL;

CREATE OR REPLACE TYPE BODY T\_Person AS

MEMBER FUNCTION Age RETURN NUMBER IS

BEGIN

RETURN TRUNC((SYSDATE - SELF.p\_birthdate)/365.25);

END Age;

END;

CREATE OR REPLACE TYPE T\_Manager UNDER T\_Person(

m\_id NUMBER,

m\_appointed DATE,

m\_total\_experience NUMBER

);

CREATE TABLE Managers OF T\_Manager(

m\_id PRIMARY KEY

)

OBJECT IDENTIFIER IS PRIMARY KEY;

INSERT INTO Managers VALUES(T\_Manager('Jurgen', 'Klopp', TO\_DATE('16-06-1967', 'DD-MM-YYYY'), 'German', '1', TO\_DATE('08-10-2015', 'DD-MM-YYYY'), '20'));

INSERT INTO Managers VALUES(T\_Manager('Xavi', 'Hernandez', TO\_DATE('25-01-1980', 'DD-MM-YYYY'), 'Spanish', '2', TO\_DATE('06-11-2021', 'DD-MM-YYYY'), '2'));

INSERT INTO Managers VALUES(T\_Manager('Carlo', 'Anchelotti', TO\_DATE('10-06-1959', 'DD-MM-YYYY'), 'Italian', '3', TO\_DATE('01-06-2021', 'DD-MM-YYYY'), '26'));

INSERT INTO Managers VALUES(T\_Manager('Ole Gunnar', 'Solskjaer', TO\_DATE('26-02-1973', 'DD-MM-YYYY'), 'Norwegian', '4', TO\_DATE('19-12-2018', 'DD-MM-YYYY'), '10'));

INSERT INTO Managers VALUES(T\_Manager('Massimiliano', 'Allegri', TO\_DATE('11-08-1967', 'DD-MM-YYYY'), 'Italian', '5', TO\_DATE('28-05-2021', 'DD-MM-YYYY'), '18'));

INSERT INTO Managers VALUES(T\_Manager('Stefano', 'Pioli', TO\_DATE('20-10-1965', 'DD-MM-YYYY'), 'Italian', '6', TO\_DATE('09-10-2019', 'DD-MM-YYYY'), '18'));

SELECT VALUE(m)

FROM Managers m;

SELECT VALUE(m)

FROM Managers m

WHERE VALUE(m).m\_id = 1;

SELECT m.\*

FROM Managers m

ORDER BY m\_appointed DESC;

SELECT m.\*

FROM Managers m

WHERE m.m\_total\_experience > 15

ORDER BY m\_total\_experience DESC;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tabula Managers\_Salary ar objektu kolonu T\_Contract

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Contract AS OBJECT(

wage NUMBER,

expires NUMBER

);

CREATE TABLE Managers\_Salaries(

m\_rec\_id NUMBER,

m\_contract T\_Contract,

m\_c\_ref REF T\_Manager SCOPE IS Managers

);

INSERT INTO Managers\_Salaries VALUES('1', T\_Contract('230000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 1));

INSERT INTO Managers\_Salaries VALUES('2', T\_Contract('150000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 2));

INSERT INTO Managers\_Salaries VALUES('3', T\_Contract('200000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 3));

INSERT INTO Managers\_Salaries VALUES('4', T\_Contract('205000', '2024'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 4));

INSERT INTO Managers\_Salaries VALUES('5', T\_Contract('140000', '2025'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 5));

INSERT INTO Managers\_Salaries VALUES('6', T\_Contract('135000', '2022'), (SELECT REF(m) FROM Managers m WHERE m.m\_id = 6));

SELECT m.\*

FROM Managers\_Salaries m;

SELECT m.m\_rec\_id, m.m\_contract.wage, m.m\_contract.expires

FROM Managers\_Salaries m;

SELECT m.m\_c\_ref.p\_name, m.m\_c\_ref.p\_surname, m.m\_contract.wage, m.m\_contract.expires

FROM Managers\_Salaries m;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Objektu tabula Players ar saliktu objektu T\_Person(Age()) <- T\_Player(T\_Contract)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Player UNDER T\_Person(

p\_id NUMBER,

p\_position VARCHAR2(30),

p\_contract T\_Contract

);

CREATE TABLE Players OF T\_Player;

INSERT INTO Players VALUES(T\_Player('Cristiano', 'Ronaldo', TO\_DATE('05-02-1985', 'DD-MM-YYYY'), 'Portugese', '1', 'Forward', T\_Contract('430000', '2023')));

INSERT INTO Players VALUES(T\_Player('Paul', 'Pogba', TO\_DATE('15-03-1993', 'DD-MM-YYYY'), 'French', '2', 'Midfielder', T\_Contract('340000', '2022')));

INSERT INTO Players VALUES(T\_Player('Bruno', 'Fernandes', TO\_DATE('08-09-1994', 'DD-MM-YYYY'), 'Portugese', '3', 'Midfielder', T\_Contract('210000', '2025')));

INSERT INTO Players VALUES(T\_Player('Virgil', 'van Dijk', TO\_DATE('08-07-1991', 'DD-MM-YYYY'), 'Dutch', '4', 'Defender', T\_Contract('260000', '2025')));

INSERT INTO Players VALUES(T\_Player('Mohamed', 'Salah', TO\_DATE('15-06-1992', 'DD-MM-YYYY'), 'Egyptian', '5', 'Forward', T\_Contract('235000', '2023')));

INSERT INTO Players VALUES(T\_Player('Allison', 'Becker', TO\_DATE('02-10-1992', 'DD-MM-YYYY'), 'Brazilian', '6', 'Goalkeeper', T\_Contract('175000', '2027')));

INSERT INTO Players VALUES(T\_Player('Zlatan', 'Ibrahimovic', TO\_DATE('03-10-1981', 'DD-MM-YYYY'), 'Swedish', '7', 'Forward', T\_Contract('205000', '2022')));

INSERT INTO Players VALUES(T\_Player('Olivier', 'Giroud', TO\_DATE('30-09-1986', 'DD-MM-YYYY'), 'French', '8','Forward', T\_Contract('100000', '2023')));

INSERT INTO Players VALUES(T\_Player('Fikayo', 'Tomori', TO\_DATE('19-12-1997', 'DD-MM-YYYY'), 'English', '9', 'Defender', T\_Contract('60000', '2025')));

INSERT INTO Players VALUES(T\_Player('Wojciech', 'Szczesny', TO\_DATE('18-04-1990', 'DD-MM-YYYY'), 'Polish', '10', 'Goalkeeper', T\_Contract('232000', '2024')));

INSERT INTO Players VALUES(T\_Player('Leonardo', 'Bonucci', TO\_DATE('01-05-1987', 'DD-MM-YYYY'), 'Italian', '11', 'Defender', T\_Contract('211000', '2024')));

INSERT INTO Players VALUES(T\_Player('Manuel', 'Locatelli', TO\_DATE('08-01-1998', 'DD-MM-YYYY'), 'Italian', '12', 'Midfielder', T\_Contract('117000', '2026')));

INSERT INTO Players VALUES(T\_Player('Karim', 'Benzema', TO\_DATE('19-08-1987', 'DD-MM-YYYY'), 'French', '13', 'Forward', T\_Contract('320000', '2023')));

INSERT INTO Players VALUES(T\_Player('Vinicius', 'Jr', TO\_DATE('12-07-2000', 'DD-MM-YYYY'), 'Brazilian', '14', 'Forward', T\_Contract('215000', '2025')));

INSERT INTO Players VALUES(T\_Player('Luka', 'Modric', TO\_DATE('09-09-1985', 'DD-MM-YYYY'), 'Croatian', '15', 'Midfielder', T\_Contract('322000', '2022')));

INSERT INTO Players VALUES(T\_Player('Gerard', 'Pique', TO\_DATE('02-02-1987', 'DD-MM-YYYY'), 'Spanish', '16', 'Defender', T\_Contract('120000', '2024')));

INSERT INTO Players VALUES(T\_Player('Marc-Andre', 'ter Stegen', TO\_DATE('30-04-1992', 'DD-MM-YYYY'), 'German', '17', 'Goalkeeper', T\_Contract('165000', '2025')));

INSERT INTO Players VALUES(T\_Player('Frenkie', 'de Jong', TO\_DATE('12-05-1997', 'DD-MM-YYYY'), 'Dutch', '18', 'Midfielder', T\_Contract('415000', '2024')));

SELECT VALUE(p)

FROM Players p;

SELECT p.\*

FROM Players p;

SELECT OBJECT\_ID, p.\*

FROM Players p;

SELECT VALUE(p).p\_name, VALUE(p).p\_surname, VALUE(p).p\_contract.wage, VALUE(p).p\_contract.expires

FROM Players p;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Objektu tabula Stats ar vienkarso objektu T\_Stat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Stat AS OBJECT(

stat\_id NUMBER,

games\_played NUMBER,

goals\_scored NUMBER,

goals\_conceded NUMBER,

ORDER MEMBER FUNCTION compare\_goals(c T\_Stat) RETURN INTEGER

);

CREATE OR REPLACE TYPE BODY T\_Stat AS

ORDER MEMBER FUNCTION compare\_goals(c T\_Stat) RETURN INTEGER IS

BEGIN

IF SELF.goals\_scored < c.goals\_scored THEN RETURN -1;

ELSIF SELF.goals\_scored > c.goals\_scored THEN RETURN 1;

ELSE RETURN 0;

END IF;

END compare\_goals;

END;

CREATE TABLE Stats OF T\_Stat(

stat\_id PRIMARY KEY

)

OBJECT IDENTIFIER IS PRIMARY KEY;

INSERT INTO Stats VALUES(T\_Stat('1', '13', '21', '22'));

INSERT INTO Stats VALUES(T\_Stat('2', '13', '39', '11'));

INSERT INTO Stats VALUES(T\_Stat('3', '14', '30', '18'));

INSERT INTO Stats VALUES(T\_Stat('4', '14', '18', '16'));

INSERT INTO Stats VALUES(T\_Stat('5', '14', '34', '15'));

INSERT INTO Stats VALUES(T\_Stat('6', '14', '23', '16'));

SELECT VALUE(s)

FROM Stats s;

SELECT s.\*

FROM Stats s;

SELECT VALUE(st).stat\_id, t.team\_name, VALUE(st).compare\_goals(c.mid)

FROM Stats st, Teams t, (SELECT VALUE(stat) AS mid

FROM Stats stat WHERE VALUE(stat).stat\_id = '1')c

WHERE team\_id = stat\_id;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tabula Teams ar objektu kolekciju Players\_ref

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE T\_Player\_ref AS OBJECT(

p\_ref\_id NUMBER,

p\_ref\_info REF T\_Player

);

CREATE OR REPLACE TYPE Players\_ref AS TABLE OF T\_Player\_ref;

CREATE TABLE Teams(

team\_id NUMBER,

team\_name VARCHAR2(40),

team\_country VARCHAR2(30),

foundation\_year NUMBER(4),

p\_ref Players\_ref,

m\_ref REF T\_Manager SCOPE IS Managers,

s\_ref REF T\_Stadium SCOPE IS Stadiums,

stat\_ref REF T\_Stat SCOPE IS Stats

)

NESTED TABLE p\_ref STORE AS Squad;

DECLARE

TYPE T\_Player\_arr IS VARRAY(18) OF REF T\_Player;

Players\_arr T\_Player\_arr := T\_Player\_arr();

BEGIN

Players\_arr.EXTEND(18);

FOR i IN 1..18 LOOP

SELECT REF(p) INTO Players\_arr(i) FROM Players p WHERE p.p\_id = (i);

END LOOP;

INSERT INTO Teams VALUES('1', 'Manchester United', 'England', '1878', Players\_ref(T\_Player\_ref('1', Players\_arr(1)),

T\_Player\_ref('2', Players\_arr(2)),

T\_Player\_ref('3', Players\_arr(3))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 4),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 3),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 1));

INSERT INTO Teams VALUES('2', 'Liverpool FC', 'England', '1892', Players\_ref(T\_Player\_ref('4', Players\_arr(4)),

T\_Player\_ref('5', Players\_arr(5)),

T\_Player\_ref('6', Players\_arr(6))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 1),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 6),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 2));

INSERT INTO Teams VALUES('3', 'AC Milan', 'Italy', '1899', Players\_ref(T\_Player\_ref('7', Players\_arr(7)),

T\_Player\_ref('8', Players\_arr(8)),

T\_Player\_ref('9', Players\_arr(9))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 6),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 1),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 3));

INSERT INTO Teams VALUES('4', 'Juventus', 'Italy', '1897', Players\_ref(T\_Player\_ref('10', Players\_arr(10)),

T\_Player\_ref('11', Players\_arr(11)),

T\_Player\_ref('12', Players\_arr(12))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 5),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 4),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 4));

INSERT INTO Teams VALUES('5', 'Real Madrid', 'Spain', '1902', Players\_ref(T\_Player\_ref('13', Players\_arr(13)),

T\_Player\_ref('14', Players\_arr(14)),

T\_Player\_ref('15', Players\_arr(15))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 3),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 2),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 5));

INSERT INTO Teams VALUES('6', 'FC Barcelona', 'Spain', '1899', Players\_ref(T\_Player\_ref('16', Players\_arr(16)),

T\_Player\_ref('17', Players\_arr(17)),

T\_Player\_ref('18', Players\_arr(18))),

(SELECT REF(m) FROM Managers m WHERE m.m\_id = 2),

(SELECT REF(s) FROM Stadiums s WHERE s.st\_id = 5),

(SELECT REF(st) FROM Stats st WHERE st.stat\_id = 6));

END;

SELECT t.\*

FROM Teams t;

SELECT t.team\_name, t.stat\_ref.goals\_scored, t.stat\_ref.goals\_conceded, (t.stat\_ref.goals\_scored - t.stat\_ref.goals\_conceded) AS goal\_difference

FROM Teams t;

SELECT t.team\_name, t.s\_ref.st\_name, t.m\_ref.p\_name, t.m\_ref.p\_surname

FROM Teams t;

SELECT t.team\_name, t.m\_ref.p\_name, t.m\_ref.p\_surname, t.m\_ref.Age()

FROM Teams t;

SELECT t.team\_name, VALUE(p)

FROM Teams t, TABLE(t.p\_ref) p;

SELECT t.team\_name, DEREF(p.p\_ref\_info).p\_name, DEREF(p.p\_ref\_info).p\_surname

FROM Teams t, TABLE(t.p\_ref) p;

SELECT t.team\_name, DEREF(p.p\_ref\_info).p\_name, DEREF(p.p\_ref\_info).p\_surname, DEREF(p.p\_ref\_info).p\_nationality

FROM Teams t, TABLE(t.p\_ref) p

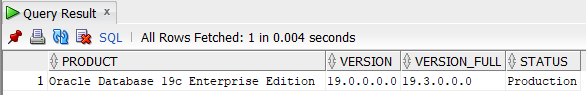
WHERE DEREF(p.p\_ref\_info).p\_nationality IN ('French', 'Portugese')

ORDER BY team\_name;

## **Darba poligona apraksts**

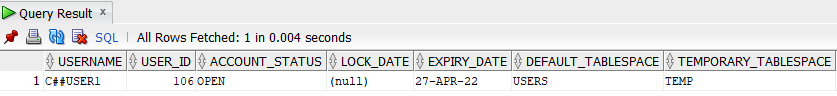
Vispirms, kad es instalēju Oracle SQL Developer es uz sava datora izveidoju jauno Windows profili, lai to izmantotu tīri datu bāzes sistēmas nolūkam. Pēc instalācijas, pirmo reizi Developer rīkā es pieslēdzos ar SYS lietotāju, izveidojot tām jauno pieslēgumu.

**SELECT \* FROM PRODUCT\_COMPONENT\_VERSION;**



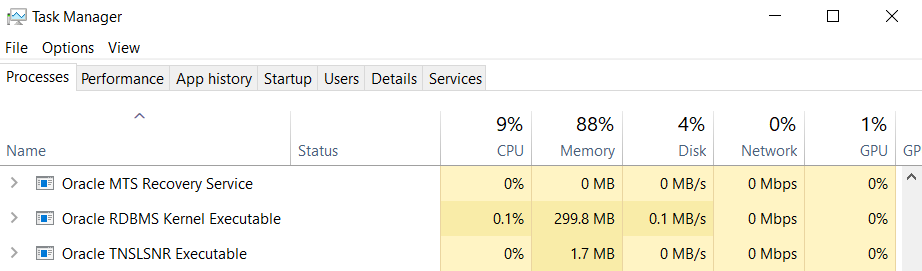
Pēc tām, es izveidoju jauno “common” lietotāju c##user1, lai ikdienā neizmantotu lietotāju ar administratora tiesībām. Šīm lietotājām arī izveidojot jaunu pieslēgumu, to es arī un izmantoju darbu izpildes laikā.

**SELECT \* FROM user\_users;**



**Oracle Procesi**





**Oracle Servisi**

