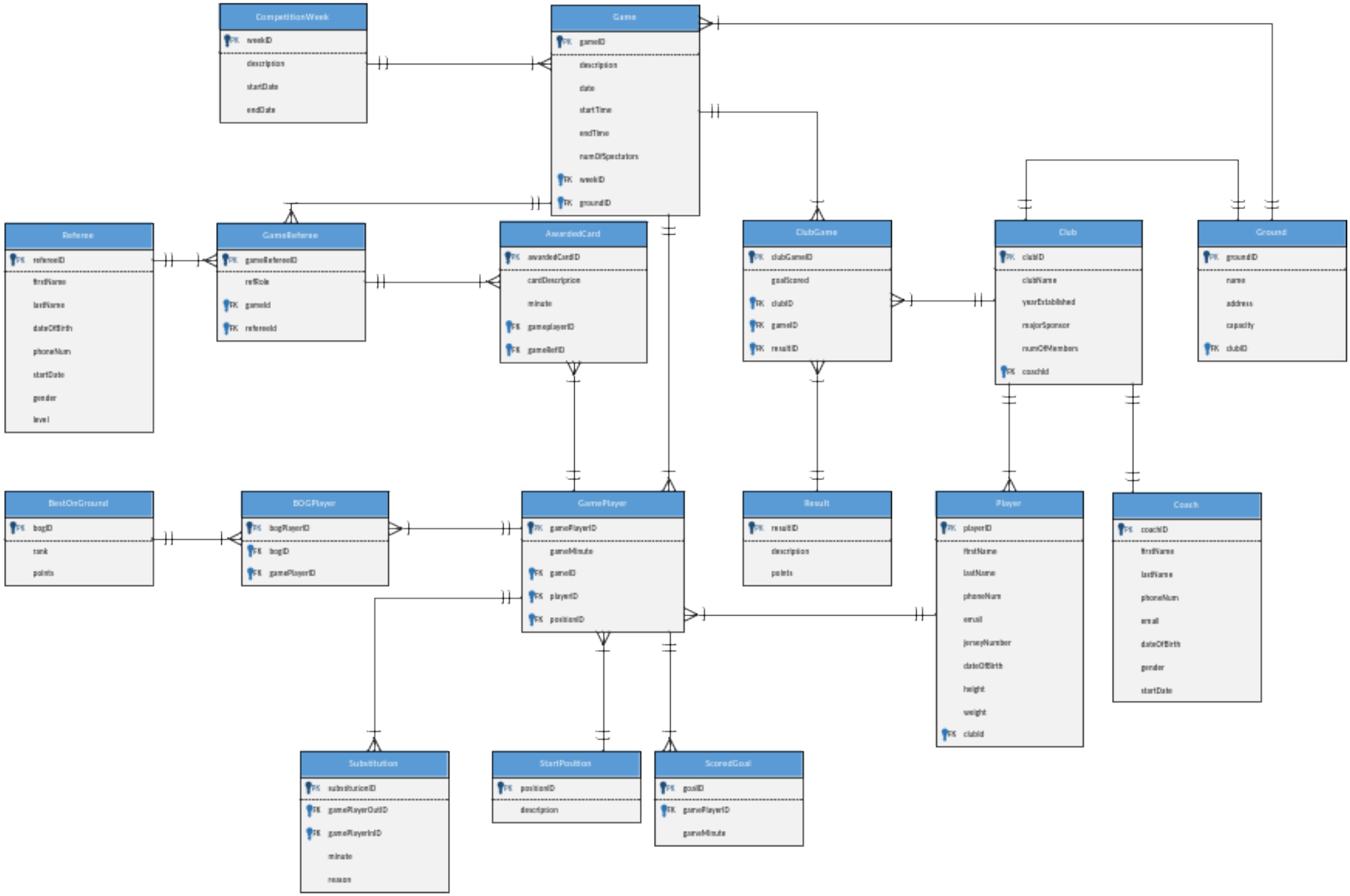
1. Design / Conceptual DB

1.1 Entity Relationship Diagram


CompetitionWeek
PK weekID
description
startDate
endDate
Game
PK gameID
description
date
startTime
endTime
numOfSpectators
FK weekID
FK groundID
Referee
PK refereeID
firstName
lastName
dateOfBirth
phoneNum
startDate
gender
level
GameReferee
PK gameRefereeID
refRole
FK gameId
FK refereeId
AwardedCard
PK awardedCardID
cardDescriptice
minute
FK gameplayerID
FK gameRefID
ClubGame
PK clubGameID
goalScored
FK clubID
FK gameID
FK resultID
Club
PK clubID
clubName
yearEstablished
majorSponsor
numOfMembers
FK coachId
Ground
PK groundID
name
address
capacity
FK clubID
BestOnGround
PK bogID
rank
points
BOGPlayer
PK bogPlayerID
FK bogID
FK gamePlayerID
GamePlayer
PK gamePlayerID
gameMinute
FK gameID
FK playerID
FK positionID
Result
PK resultID
description
points
Player
PK playerID
firstName
lastName
phoneNum
email
jerseyNumber
dateOfBirth
height
weight
FK clubId
Coach
PK coachID
firstName
lastName
phoneNum
email
dateOfBirth
gender
startDate
Substitution
PK substituticeID
FK gamePlayerOutID
FK gamePlayerInID
minute
reason
StartPosition
PK positionID
description
ScoredGoal
PK goalID
FK gamePlayerID
gameMinute

1.2 Business Rules

1. CompetitionWeek to Game

1a. A CompetitionWeek includes one or more than one Game.

1b. A Game is played in one and only one CompetitionWeek.

2. Game and GameReferee

2a. A Game appears in one or more than one GameReferee.

2b. A GameReferee supervises one and only one Game.

3. Referee and GameReferee

3a. A Referee appears in one or more than one GameReferee

3b. A GameReferee records one and only one Referee.

4. Ground and Game

4a. A Ground hosts one or more than one Game.

4b. A Game takes place at one and only one Ground.

5. Game and ClubGame

5a. A game is recorded in one or more than one ClubGame.

5b. A ClubGame records one and only one Game.

6. Club and ClubGame

6a. A ClubGame records one and only one Club.

6b. A Club plays in one or more than one ClubGame.

7. Club and Ground

7a. A Club plays in one and only one Ground.

7b. A Ground hosts one and only one Club.

8. Game and Gameplayer

8a. A GamePlayer contains one and only one game

8b. A Game is recorded in one or more than one GamePlayer.

9. GamePlayer and AwardedCard

9a. A GamePlayer gets one or more than one AwardedCard.

9b. An AwardedCard records one and only one GamePlayer.

10. GameReferee to AwardedCard

10a. A GameReferee appears in one or more than one AwardedCard.

10b. An AwardedCard is issued by one and only one GameReferee.

11. BestOnGround and BOGPlayer

11a. A BestOnGround appears in one or more than one BOGPlayer.

11b. A BOGPlayer records one and only one BestOnGround.

12. GamePlayer and BOGPlayer

12a. A GamePlayer appears in one or more than one BOGPlayer.

12b. A BOGPlayer records one and only one GamePlayer.

13. Result and ClubGame

13a. A Result records one or more than one ClubGame.

13b. A ClubGame has one and only one Result.

14. Club and Player

14a. A Club has one or more than one Player.

14b. A Player participates in one and only one club.

15. Club and Coach

15a. A Club has one and only one Coach.

15b. A Coach works for one and only one Club.

16. Player and GamePlayer

16a. A Player participates in one or more than one GamePlayer.

16b. A GamePlayer records one and only one Player.

17. GamePlayer and ScoredGoal

17a. A GamePlayer gets one or more than one ScoredGoals.

17b. A ScoredGoal is recorded in one and only one GamePlayer.

18. StartPosition and GamePlayer

18a. A StartPosition is assigned to one or more than one GamePlayers.

18b. A GamePlayer has one and only one StartPosition.

19. GamePlayer and Substitution

19a. A GamePlayer participates in one or more than one Substitution.

19b. A Substitution appears in one and only one GamePlayer.

2. Schema / Logical DB Schema

2.1 DDL-Schema

TABLE	COLUMNS
COACH	(**coachID [PK]**, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)
COMPETITIONWEEK	(**weekID [PK]**, description, startDate, endDate)
REFEREE	(**refereeID [PK]**, firstName, lastName, dateOfBirth, phoneNum, starDate, gender, refLevel)
STARTPOSITION	(**positionID [PK]**, description)
BESTONGROUND	(**bogID [PK]**, rank, points)
RESULT	(**resultID [PK]**, description, points)
CLUB	(**clubID [PK]**, clubName, yearEstablished, majorSponsor, numOfMembers, *coachID [FK])*
PLAYER	(**playerID[PK]**, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, *clubID[FK])*
GROUND	(**groundID [PK]**, name, address, capacity, *clubID [FK])*
GAME	(**gameID [PK]**, description, gameDate, startTime, endTime, numOfSpectators *weekID [FK], groundID [FK])*
CLUBGAME	(**clubGameID [PK]**, *gameID [FK]*, *clubID [FK],* goalScored, *resultID [FK])*
GAMEREFEREE	(**gameRefereeID[PK]**, refRole, *gameID [FK], refereeID [FK])*
GAMEPLAYER	(**gamePlayerID [PK]**, *gameID [FK], playerID [FK], positionID [FK],* gameMinute
BOGPLAYER	(**bogPlayerID [PK]**, *bogID [FK], gamePlayerID [FK]*)
SUBSTITUTION	(**substitutionID[PK]**, *gamePlayerOutID [FK], gamePlayerInID [FK],* minute, reason)
AWARDEDCARD	(**awardedCardID [PK]**, cardDescription, minute, *gamePlayerID [FK], gameRefID [FK]*)
SCOREDGOAL	(**goalID [PK]**, *gamePlayerID [FK]*, gameMinute)

KEY: **Bold Underline** = PK, *Underline Italic* = FK

2.2 Data Dictionary

Format details: FONT=Calibri, size=8, color=black. PARAGRAPH space before=0, space after=0. TABLE bg-color=RGB/204,102,0 border-color=RGB/153,51,0.

TableName	Alias	Column Name	Description	Format	Opt/Require	PK Name	PK Column Ref	FK Name	FK Column Ref	Table Ref	Other C.Name(S)
Coach	ch	coachID	Coach's unique ID and PK in Coach	CHAR (4)	NOT NULL	Coach_PK	coachID				
		firstName	Coach's first name	VARCHAR2 (25)	NOT NULL						
		lastName	Coash's last name	VARCHAR2 (25)	NOT NULL						
		phoneNum	Coach's phone number	CHAR (10)							
		email	Coach's email address	VARCHAR2(30)							
		dateOfBirth	Date of birth of a coach	DATE							
		gender	Coach's gender	CHAR (1)							
		startDate	The coach's employment start date	DATE							
Competition Week	cw	weekID	Unique ID of competition week and PK in CompetitionWeek	CHAR (4)	NOT NULL	CompetitionWeek_PK	weekID				
		description	Description of competition week	VARCHAR2(25)	NOT NULL						
		startDate	Start date of competition week	DATE	NOT NULL						
		endDate	End date of competition week	DATE	NOT NULL						
Referee	ref	refereeID	Unique identification of referee and PK in Referee	CHAR (4)	NOT NULL	Referee_PK	refereeID				
		firstName	Referee's first name	VARCHAR2(25)	NOT NULL						
		lastName	Referee's last name	VARCHAR2(25)	NOT NULL						
		dateOfBirth	Referee's date of birth	DATE							
		phoneNum	Referee's phone number	CHAR(10)							
		startDate	Referee's employment start date	DATE							
		gender	The gender of the referee	CHAR(1)							

			(M/W)								
		refLevel	The qualification of the referee (1-4)	CHAR (1)							
StartPosition	sp	positionID	Unique ID of a pitch start position and PK in StartPosition	CHAR(4)	NOT NULL	StartPosition_PK	positionID				
		description	Name of the pitch start position	VARCHAR(40)	NOT NULL						
BestOnGround	bog	bogID	Unique ID and PK in BestOnGround. A BestOnGround record holds information about performance scoring data such as rank and associated points.	CHAR(3)	NOT NULL	BestOnGround_PK	bogID				
		rank	Rank associated with the record	CHAR(1)							
		Points	Points associated with the record	NUMBER(1,0)							
Result	rs	resultID	The unique ID of a result	CHAR(4)	NOT NULL	Result_PK	resultID				
		description	The type of result (HomeWin, AwayWin, HomeDraw, AwayDraw, HomeLoss, AwayLoss)	VARCHAR(15)	NOT NULL						
		points	Points associated with a result	NUMBER(1,0)	NOT NULL						
Club	cl	clubID	Unique ID of a club and PK in CLub	CHAR (4)	NOT NULL	Club_PK	clubID				
		clubName	CLub's name	VARCHAR 2 (25)	NOT NULL						
		yearEstablished	Club's establishment year	NUMBER (4,0)							
		majorSponsor	Club's major sponsor	VARCHAR2 (25)							
		numOfMembers	Club's total member	NUMBER (5,0)							
		coachID	CLubs's coach and FK from Coach	CHAR (4)	NOT NULL			Coach_FK	CoachID	Coach	

Player	pl	playerID	Unique ID of a PLayer	CHAR (5)	NOT NULL	Player_PK	playerID				
		firstName	Player's first name	VARCHAR2 (30)	NOT NULL						
		lastName	Player's last name	VARCHAR2 (30)	NOT NULL						
		phoneNum	Player's phone number	CHAR (10)							
		email	Player's email address	VARCHAR2 (30)							
		jerseyNumber	Jersey/shirt number	VARCHAR2 (2)							
		dateOfBirth	Player's date of birth	DATE							
		height	Player's height (cm)	NUMBER(3,0)							
		weight	Player's weight (kg)	NUMBER (3,0)							
		clubID	The club a player plays for and FK from Club	CHAR (4)	NOT NULL			Player_Club_FK	clubID	Club	
Ground	grn	groundID	Unique ID of a ground	CHAR (4)	NOT NULL	Ground_PK	groundID				
		name	Ground's name	VARCHAR2 (30)	NOT NULL						
		address	Ground's address	VARCHAR 2 (40)	NOT NULL						
		capacity	Ground capacity	NUMBER (5,0)	NOT NULL						
		clubID	Club who trains on the ground and FK from Club	VARCHAR (4)	NOT NULL			Ground_Club_FK	clubID	Club	
Game	gm	gameID	The unique ID of a game	CHAR (4)	NOT NULL	Game_PK	gameID				
		description	Game description (fixture)	VARCHAR (15)	NOT NULL						
		gameDate	Date a game takes place on	DATE	NOT NULL						
		startTime	The start time of a game.	TIME	NOT NULL						
		endTime	The end time of the game.	TIME	NOT NULL						
		numOfSpectators	Number of people watching the game	NUMBER (5,0)							
		weekID	Week which a game plays in and	CHAR (4)	NOT NULL			Game_	weekID	CompetitionWeek	

			FK from Week					CompetitionWeek_FK			
		groundID	Ground a game play on.	CHAR (4)	NOT NULL			Game_ Ground_FK	groundID	Ground	
ClubGame	cg	clubGameID	The unique ID of a record kept about a game and its participating club. It is also a PK in ClubGame	CHAR (5)	NOT NULL	clubGame_PK	clubGameID				
		gameID	ID of a game and FK from Game	CHAR (4)	NOT NULL			clubGame_Game_FK	gameID	Game	
		clubID	ID of a club and FK from Club	CHAR (4)	NOT NULL			clubGame_Club_FK	clubID	Club	
		goalScored	The number of goals scored in the game	NUMBER (2,0)	NOT NULL						
		resultID	Result ID and FK from Result	CHAR (4)	NOT NULL			clubGame_Result_FK	resultID	Result	
GameReferee	gr	gameRefereeID	Unique ID of a record kept about a game its associated Referee	CHAR (6)	NOT NULL	GameReferee_PK	GameRefereeID				
		refRole	The role of referee in the game	VARCHAR2 (15)	NOT NULL						
		gameID	The game which the referee refereed in and FK from Game	CHAR (4)	NOT NULL			GR_Game_FK	gameID	Game	
		refereeID	The referee associated with the game and FK from Referee	CHAR (5)	NOT NULL			GR_Referee_FK	refereeID	Referee	
GamePlayer	gp	gamePlayerID	Unique ID of a record of each player's participation in a game and PK in GamePlayer	CHAR(6)	NOT NULL	GamePlayer_PK	gamePlayerID				
		gameID	The game the player played in and FK from Game	CHAR(5)	NOT NULL			GamePlayer_ Game_FK	gameID	Game	
		playerID	The player who played in the game and FK from Player	CHAR(5)	NOT NULL			GamePlayer_ Player_FK	playerID	Player	
		positionID	Starting position of the player and FK from StartPosition	CHAR(5)	NOT NULL			GamePlayer_ Position_FK	positionID	StartPosition	

		gameMinute	Minutes the player played for in the game	NUMBER (3,0)	NOT NULL						
Substitution	sub	substitutionID	Unique ID of each substitution made in a game and PK in Substitution	CHAR (6)	NOT NULL	Substitution_PK	substitutionID				
		gamePlayerOutID	The ID of a GamePlayer record where its associated player got substituted out and FK from GamePlayer	CHAR (6)	NOT NULL			GamePlayerOut_ID_FK	gamePlayerID	GamePlayer	
		gamePlayerInID	The ID of a GamePlayer record where its associated player got substituted in and FK from GamePlayer	CHAR (6)	NOT NULL			GamePlayerIn_ID_FK	gamePlayerID	GamePlayer	
		minute	Minute the substitution happened at	NUMBER (3,0)	NOT NULL						
		reason	The reason for substitution	VARCHAR2 (30)							
AwardedCard	ac	awardedCardID	Unique ID of each awarded card PK in AwardedCard	CHAR(5)	NOT NULL	AwardedCard_PK	awardedCardID				
		cardDescription	Type of card (YellowCard, RedCard)	VARCHAR2 (6)	NOT NULL						
		minute	Minute of the game when the card was awarded	NUMBER(3,0)	NOT NULL						
		gamePlayerID	An ID of GamePlayer record where its player was awarded a card and FK from GamePlayer	CHAR (6)	NOT NULL			AwardedCard_ GamePlayer_FK	gamePlayerID	GamePlayer	
		gameRefereeID	An ID of a record of GameReferee associated referee in the game awarded the card.	CHAR (6)	NOT NULL			AwardedCard_ GameReferee_FK	gameRefereeID	GameReferee	
ScoredGoal	sg	goalID	A unique ID of a goal and PK in ScoredGoal	CHAR (5)	NOT NULL	ScoredGoal_PK	scoredGoalID				
		gamePlayerID	An ID of a record of GamePlayer	CHAR (6)	NOT NULL			ScoredGoal_	gamePlayerID	GamePlayer	

			that links a game and the player who scored the goal.					GamePlayer_FK			
		gameMinute	The minute at which the goal was scored	NUMBER (3,0)	NOT NULL						
BogPlayer	bp	bogPlayerID	Unique ID of a record kept identifying a player who was nominated a Best on Ground.	CHAR(6)	NOT NULL	BOGPlayer_PK	bogPlayerID				
		bogID	An ID of a record of BestOnGround and FK from BestOnGround.	CHAR(3)	NOT NULL			BogPlayer_ Bog_FK	bogID	BOG	
		gamePlayerID	An ID of a record of GamePlayer where its associated player got nominated as a best on ground and FK from GamePlayer	CHAR(6)	NOT NULL			BogPlayer_ GamePlayer_FK	bogID	GamePlayer	

3. SQL Implementation

3.1 CREATE Table SQL Syntax

Coach

The SQL Syntax:

```
CREATE TABLE Coach (
        coachID CHAR(4) NOT NULL,
        firstName VARCHAR2(25) NOT NULL,
        lastName VARCHAR2(25) NOT NULL,
        phoneNum CHAR(10),
        email VARCHAR2(30),
        dateOfBirth DATE,
        gender CHAR(1),
        startDate DATE,
        CONSTRAINT Coach_PK PRIMARY KEY (coachID)
);
```

Result of syntax in Oracle:

COACH

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
COACHID	CHAR(4)	No	-	1
FIRSTNAME	VARCHAR2(25)	No	-	-
LASTNAME	VARCHAR2(25)	No	-	-
PHONENUM	CHAR(10)	Yes	-	-
EMAIL	VARCHAR2(30)	Yes	-	-
DATEOFBIRTH	DATE	Yes	-	-
GENDER	CHAR(1)	Yes	-	-
STARTDATE	DATE	Yes	-	-

Download | Print

COACH

Table | Data | Indexes | Model | **Constraints** | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205315	Check	"COACHID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:04:33 AM	-	-
SYS_C00205316	Check	"FIRSTNAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:04:33 AM	-	-
SYS_C00205317	Check	"LASTNAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:04:33 AM	-	-
COACH_PK	Primary	-	-	COACHID	-	ENABLED	09/23/2025 07:04:33 AM	COACH_PK	-

CompetitionWeek

The SQL Syntax:

```
CREATE TABLE CompetitionWeek (
        weekID CHAR(4) NOT NULL,
        description VARCHAR2(25) NOT NULL,
        startDate DATE NOT NULL,
        endDate DATE NOT NULL,
        CONSTRAINT CompetitionWeek_PK PRIMARY KEY (weekID)
);
```

Result of syntax in Oracle:

COMPETITIONWEEK

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
WEEKID	CHAR(4)	No	-	1
DESCRIPTION	VARCHAR2(25)	No	-	-
STARTDATE	DATE	No	-	-
ENDDATE	DATE	No	-	-

Download | Print

COMPETITIONWEEK

Table | Data | Indexes | Model | **Constraints** | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205319	Check	"WEEKID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:05:43 AM	-	-
SYS_C00205320	Check	"DESCRIPTION" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:05:43 AM	-	-
SYS_C00205321	Check	"STARTDATE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:05:43 AM	-	-
SYS_C00205322	Check	"ENDDATE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:05:43 AM	-	-
COMPETITIONWEEK_PK	Primary	-	-	WEEKID	-	ENABLED	09/23/2025 07:05:43 AM	COMPETITIONWEEK_PK	-

Referee

The SQL Syntax:

```
CREATE TABLE Referee (
        refereeID CHAR (4) NOT NULL,
        firstName VARCHAR2(25) NOT NULL,
        lastName VARCHAR2(25) NOT NULL,
        dateOfBirth DATE,
        phoneNum CHAR(10),
        startDate DATE,
        gender CHAR(1),
        refLevel CHAR(1),
        CONSTRAINT Referee_PK PRIMARY KEY (refereeID)
);
```

Result of syntax in Oracle:

GAMEREFEREE

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GAMEREFEREEID	CHAR(6)	No	-	1
REFROLE	VARCHAR2(15)	No	-	-
GAMEID	CHAR(4)	No	-	-
REFEREEID	CHAR(4)	No	-	-

Download | Print

GAMEREFEREE

Table | Data | Indexes | **Model** | **Constraints** | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205436	Check	"GAMEREFEREEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205437	Check	"REFROLE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205438	Check	"GAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205439	Check	"REFEREEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
GAMEREFEREE_PK	Primary	-	-	GAMEREFEREEID	-	ENABLED	09/23/2025 09:01:06 AM	GAMEREFEREE_PK	-
GR_GAME_FK	Foreign	-	GR_GAME_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAME)	GAMEID	NO ACTION	ENABLED	09/23/2025 09:01:06 AM	-	-
GR_REFEREE_FK	Foreign	-	GR_REFEREE_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.REFEREE)	REFEREEID	NO ACTION	ENABLED	09/23/2025 02:26:02 PM	-	-

StartPosition

The SQL Syntax:

```
CREATE TABLE StartPosition (
        positionID CHAR(4) NOT NULL,
        description VARCHAR2(25) NOT NULL,
        CONSTRAINT StartPosition_PK PRIMARY KEY (positionID)
);
```

Result of syntax in Oracle:

STARTPOSITION

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
POSITIONID	CHAR(4)	No	-	1
DESCRIPTION	VARCHAR2(25)	No	-	-

Download | Print

STARTPOSITION

Table | Data | Indexes | Model | **Constraints** | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205328	Check	"POSITIONID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:07:11 AM	-	-
SYS_C00205329	Check	"DESCRIPTION" IS NOT NULL	-	-	-	ENABLED	09/23/2025 07:07:11 AM	-	-
STARTPOSITION_PK	Primary	-	-	POSITIONID	-	ENABLED	09/23/2025 07:07:11 AM	STARTPOSITION_PK	-

BestOnGround

The SQL Syntax:

```
CREATE TABLE BestOnGround (

        bogID CHAR(3) NOT NULL,

        rank CHAR(1) NOT NULL,

        points NUMBER(3) NOT NULL,

        CONSTRAINT BestOnGround_PK PRIMARY KEY (bestOnGroundID)

);
```

Result of syntax in Oracle:

BESTONGROUND

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Depen

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
BOGID	CHAR(3)	No	-	1
RANK	CHAR(1)	No	-	-
POINTS	NUMBER(3,0)	No	-	-

Download | Print

BESTONGROUND

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205343	Check	"RANK" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:24:23 AM	-	-
SYS_C00205344	Check	"POINTS" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:24:23 AM	-	-
SYS_C00205342	Check	"BOGID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:24:23 AM	-	-
BESTONGROUND_PK	Primary	-	-	BOGID	-	ENABLED	09/23/2025 08:24:23 AM	BESTONGROUND_PK	-

Result

The SQL Syntax:

```
CREATE TABLE Result (
```

```
        resultID CHAR(4) NOT NULL,

        description VARCHAR2(15) NOT NULL,

        points NUMBER(1,0) NOT NULL,

        CONSTRAINT Result_PK PRIMARY KEY (resultID)

);
```

Result of syntax in Oracle:

RESULT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
RESULTID	CHAR(4)	No	-	1
DESCRIPTION	VARCHAR2(15)	No	-	-
POINTS	NUMBER(1,0)	No	-	-

Download | Print

RESULT

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205346	Check	"RESULTID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:32:15 AM	-	-
SYS_C00205347	Check	"DESCRIPTION" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:32:15 AM	-	-
SYS_C00205348	Check	"POINTS" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:32:15 AM	-	-
RESULT_PK	Primary	-	-	RESULTID	-	ENABLED	09/23/2025 08:32:15 AM	RESULT_PK	-

Club

```
CREATE TABLE Club (

        clubID CHAR(4) NOT NULL,

        clubName VARCHAR2(25) NOT NULL,

        yearEstablished NUMBER(4,0),

        majorSponsor VARCHAR2(25),

        numOfMembers NUMBER(5,0),

        coachID CHAR(4) NOT NULL,

        CONSTRAINT Club_PK PRIMARY KEY (clubID),

        CONSTRAINT Club_Coach_FK FOREIGN KEY (coachID) REFERENCES Coach(coachID)

);
```

Result of syntax in Oracle:

CLUB

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
CLUBID	CHAR(4)	No	-	1
CLUBNAME	VARCHAR2(25)	No	-	-
YEARESTABLISHED	NUMBER(4,0)	Yes	-	-
MAJORSPONSOR	VARCHAR2(25)	Yes	-	-
NUMOFMEMBERS	NUMBER(5,0)	Yes	-	-
COACHID	CHAR(4)	No	-	-

Download | Print

CLUB

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205350	Check	"CLUBID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:36:13 AM	-	-
SYS_C00205351	Check	"CLUBNAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:36:13 AM	-	-
SYS_C00205352	Check	"COACHID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:36:13 AM	-	-
CLUB_PK	Primary	-	-	CLUBID	-	ENABLED	09/23/2025 08:36:13 AM	CLUB_PK	-
CLUB_COACH_FK	Foreign	-	CLUB_COACH_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.COACH)	COACHID	NO ACTION	ENABLED	09/23/2025 08:36:13 AM	-	-

Player

The SQL Syntax:

```
CREATE TABLE Player (

        playerID CHAR(5) NOT NULL,

        firstName VARCHAR2(30) NOT NULL,

        lastName VARCHAR2(30) NOT NULL,

        phoneNum CHAR(10),

        email VARCHAR2(30),

        jerseyNumber VARCHAR2(2),

        dateOfBirth DATE,

        height NUMBER(3,0),

        weight NUMBER(3,0),

        clubID CHAR(4) NOT NULL,
```

```
        CONSTRAINT Player_PK PRIMARY KEY (playerID),

        CONSTRAINT Player_Club_FK FOREIGN KEY (clubID) REFERENCES Club(clubID)

);
```

Result of syntax in Oracle:

PLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
PLAYERID	CHAR(5)	No	-	1
FIRSTNAME	VARCHAR2(30)	No	-	-
LASTNAME	VARCHAR2(30)	No	-	-
PHONENUM	CHAR(10)	Yes	-	-
EMAIL	VARCHAR2(30)	Yes	-	-
JERSEYNUMBER	VARCHAR2(2)	Yes	-	-
DATEOFBIRTH	DATE	Yes	-	-
HEIGHT	NUMBER(3,0)	Yes	-	-
WEIGHT	NUMBER(3,0)	Yes	-	-
CLUBID	CHAR(4)	No	-	-

Download | Print

PLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
PLAYER_CLUB_FK	Foreign	-	PLAYER_CLUB_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.CLUB)	CLUBID	NO ACTION	ENABLED	09/23/2025 08:39:14 AM	-	-
SYS_C00205359	Check	"PLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:14 AM	-	-
SYS_C00205360	Check	"FIRSTNAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:14 AM	-	-
SYS_C00205361	Check	"LASTNAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:14 AM	-	-
SYS_C00205362	Check	"CLUBID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:14 AM	-	-
PLAYER_PK	Primary	-	-	PLAYERID	-	ENABLED	09/23/2025 08:39:14 AM	PLAYER_PK	-

Ground

The SQL Syntax:

```
CREATE TABLE Ground (

        groundID CHAR(4) NOT NULL,

        name VARCHAR2(30) NOT NULL,
```

```
        address VARCHAR2(40) NOT NULL,

        capacity NUMBER(5,0) NOT NULL,

        clubID CHAR(4) NOT NULL,

        CONSTRAINT Ground_PK PRIMARY KEY (groundID),

        CONSTRAINT Ground_Club_FK FOREIGN KEY (clubID) REFERENCES Club(clubID)

);
```

Result of syntax in Oracle:

GROUND

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GROUNDID	CHAR(4)	No	-	1
NAME	VARCHAR2(30)	No	-	-
ADDRESS	VARCHAR2(25)	No	-	-
CAPACITY	NUMBER(5,0)	No	-	-
CLUBID	CHAR(4)	No	-	-

Download | Print

GROUND

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205365	Check	"GROUNDID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:49 AM	-	-
SYS_C00205366	Check	"NAME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:49 AM	-	-
SYS_C00205367	Check	"ADDRESS" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:49 AM	-	-
SYS_C00205368	Check	"CAPACITY" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:49 AM	-	-
SYS_C00205369	Check	"CLUBID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:39:49 AM	-	-
GROUND_PK	Primary	-	-	GROUNDID	-	ENABLED	09/23/2025 08:39:49 AM	GROUND_PK	-
GROUND_CLUB_FK	Foreign	-	GROUND_CLUB_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.CLUB)	CLUBID	NO ACTION	ENABLED	09/23/2025 08:39:49 AM	-	-

Game

The SQL Syntax:

```
CREATE TABLE Game (

        gameID CHAR(4) NOT NULL,
```

```
        description VARCHAR2(15) NOT NULL,

        gameDate DATE NOT NULL,

        startTime DATE NOT NULL,

        endTime DATE NOT NULL,

        weekID CHAR(4) NOT NULL,

        groundID CHAR(4) NOT NULL,

        CONSTRAINT Game_PK PRIMARY KEY (gameID),

        CONSTRAINT Game_CompetitionWeek_FK FOREIGN KEY (weekID) REFERENCES
CompetitionWeek(weekID),

        CONSTRAINT Game_Ground_FK FOREIGN KEY (groundID) REFERENCES Ground(groundID)

);
```

** The Game table was later updated to add a new column called ***numOfSpectators*** **

```
ALTER TABLE Game ADD (numOfSpectators NUMBER(5));
```

Result of syntax in Oracle:

GAME

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GAMEID	CHAR(4)	No	-	1
DESCRIPTION	VARCHAR2(15)	No	-	-
GAMEDATE	DATE	No	-	-
STARTTIME	DATE	No	-	-
ENDTIME	DATE	No	-	-
WEEKID	CHAR(4)	No	-	-
GROUNDID	CHAR(4)	No	-	-

Download | Print

GAME

Table | Data | Indexes | Model | **Constraints** | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
GAME_COMPETITIONWEEK_FK	Foreign	-	GAME_COMPETITIONWEEK_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.COMPETITIONWEEK)	WEEKID	NO ACTION	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205393	Check	"DESCRIPTION" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205394	Check	"GAMEDATE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205395	Check	"STARTTIME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205396	Check	"ENDTIME" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205397	Check	"WEEKID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205398	Check	"GROUNDID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-
GAME_PK	Primary	-	-	GAMEID	-	ENABLED	09/23/2025 08:55:15 AM	GAME_PK	-
GAME_GROUND_FK	Foreign	-	GAME_GROUND_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GROUND)	GROUNDID	NO ACTION	ENABLED	09/23/2025 08:55:15 AM	-	-
SYS_C00205392	Check	"GAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 08:55:15 AM	-	-

ClubGame

The SQL Syntax:

```
CREATE TABLE ClubGame (
        clubGameID CHAR(5) NOT NULL,
        gameID CHAR(4) NOT NULL,
        clubID CHAR(4) NOT NULL,
        goalScored NUMBER(2,0) NOT NULL,
        resultID CHAR(4) NOT NULL,
        CONSTRAINT clubGame_PK PRIMARY KEY (clubGameID),
        CONSTRAINT clubGame_game_FK FOREIGN KEY (gameID) REFERENCES Game(gameID),
        CONSTRAINT clubGame_club_FK FOREIGN KEY (clubID) REFERENCES Club(clubID),
        CONSTRAINT clubGame_result_FK FOREIGN KEY (resultID) REFERENCES Result(resultID)
);
```

Result of syntax in Oracle:

CLUBGAME

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
CLUBGAMEID	CHAR(4)	No	-	1
GAMEID	CHAR(4)	No	-	-
CLUBID	CHAR(4)	No	-	-
GOALSCORED	NUMBER(2,0)	No	-	-
RESULTID	CHAR(4)	No	-	-

Download | Print

CLUBGAME

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205443	Check	"CLUBGAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:05:37 AM	-	-
SYS_C00205444	Check	"GAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:05:37 AM	-	-
SYS_C00205445	Check	"CLUBID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:05:37 AM	-	-
SYS_C00205446	Check	"GOALSCORED" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:05:37 AM	-	-
SYS_C00205447	Check	"RESULTID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:05:37 AM	-	-
CLUBGAME_PK	Primary	-	-	CLUBGAMEID	-	ENABLED	09/23/2025 09:05:37 AM	CLUBGAME_PK	-
CLUBGAME_GAME_FK	Foreign	-	CLUBGAME_GAME_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAME)	GAMEID	NO ACTION	ENABLED	09/23/2025 09:05:37 AM	-	-
CLUBGAME_CLUB_FK	Foreign	-	CLUBGAME_CLUB_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.CLUB)	CLUBID	NO ACTION	ENABLED	09/23/2025 09:05:37 AM	-	-
CLUBGAME_RESULT_FK	Foreign	-	CLUBGAME_RESULT_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.RESULT)	RESULTID	NO ACTION	ENABLED	09/23/2025 09:05:37 AM	-	-

GameReferee

The SQL Syntax:

```
CREATE TABLE GameReferee (
        gameRefereeID CHAR(6) NOT NULL,
        refRole VARCHAR2(15) NOT NULL,
        gameID CHAR(4) NOT NULL,
        refereeID CHAR(4) NOT NULL,
        CONSTRAINT GameReferee_PK PRIMARY KEY (gameRefereeID),
        CONSTRAINT GR_Game_FK FOREIGN KEY (gameID) REFERENCES Game(gameID),
        CONSTRAINT GR_Referee_FK FOREIGN KEY (refereeID) REFERENCES Referee(refereeID)
);
```

Result of syntax in Oracle:

GAMEREFEREE

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GAMEREFEREEID	CHAR(6)	No	-	1
REFROLE	VARCHAR2(15)	No	-	-
GAMEID	CHAR(4)	No	-	-
REFEREEID	CHAR(4)	No	-	-

Download | Print

GAMEREFEREE

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205436	Check	"GAMEREFEREEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205437	Check	"REFROLE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205438	Check	"GAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
SYS_C00205439	Check	"REFEREEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:01:06 AM	-	-
GAMEREFEREE_PK	Primary	-	-	GAMEREFEREEID	-	ENABLED	09/23/2025 09:01:06 AM	GAMEREFEREE_PK	-
GR_GAME_FK	Foreign	-	GR_GAME_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAME)	GAMEID	NO ACTION	ENABLED	09/23/2025 09:01:06 AM	-	-
GR_REFEREE_FK	Foreign	-	GR_REFEREE_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.REFEREE)	REFEREEID	NO ACTION	ENABLED	09/23/2025 09:01:06 AM	-	-

GamePlayer

The SQL Syntax:

```
CREATE TABLE GamePlayer (

        gamePlayerID CHAR(6) NOT NULL,

        gameID CHAR(4) NOT NULL,

        playerID CHAR(5) NOT NULL,

        positionID CHAR(4) NOT NULL,

        gameMinute NUMBER(3,0) NOT NULL,

        CONSTRAINT GamePlayer_PK PRIMARY KEY (gamePlayerID),

        CONSTRAINT GamePlayer_Game_FK FOREIGN KEY (gameID) REFERENCES Game(gameID),

        CONSTRAINT GamePlayer_Player_FK FOREIGN KEY (playerID) REFERENCES Player(playerID),

        CONSTRAINT GamePlayer_Position_FK FOREIGN KEY (positionID) REFERENCES
StartPosition(positionID)

);
```

Result of syntax in Oracle:

GAMEPLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GAMEPLAYERID	CHAR(6)	No	-	1
GAMEID	CHAR(4)	No	-	-
PLAYERID	CHAR(5)	No	-	-
POSITIONID	CHAR(4)	No	-	-
GAMEMINUTE	NUMBER(3,0)	No	-	-

Download | Print

GAMEPLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index
GAMEPLAYER_GAME_FK	Foreign	-	GAMEPLAYER_GAME_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAME)	GAMEID	NO ACTION	ENABLED	09/23/2025 09:09:22 AM	-
SYS_C00205452	Check	"GAMEPLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:22 AM	-
SYS_C00205453	Check	"GAMEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:22 AM	-
SYS_C00205454	Check	"PLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:22 AM	-
SYS_C00205455	Check	"POSITIONID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:22 AM	-
SYS_C00205456	Check	"GAMEMINUTE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:22 AM	-
GAMEPLAYER_PK	Primary	-	-	GAMEPLAYERID	-	ENABLED	09/23/2025 09:09:22 AM	GAMEPLAYER_PK
GAMEPLAYER_PLAYER_FK	Foreign	-	GAMEPLAYER_PLAYER_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.PLAYER)	PLAYERID	NO ACTION	ENABLED	09/23/2025 09:09:22 AM	-
GAMEPLAYER_POSITION_FK	Foreign	-	GAMEPLAYER_POSITION_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.STARTPOSITION)	POSITIONID	NO ACTION	ENABLED	09/23/2025 09:09:22 AM	-

Substitution

The SQL Syntax:

```
CREATE TABLE Substitution (

        substitutionID CHAR(6) NOT NULL,

        gamePlayerOutID CHAR(6) NOT NULL,

        gamePlayerInID CHAR(6) NOT NULL,

        minute NUMBER(3,0) NOT NULL,

        reason VARCHAR2(30),
```

```
        CONSTRAINT Substitution_PK PRIMARY KEY (substitutionID),

        CONSTRAINT GamePlayerOut_ID_FK FOREIGN KEY (gamePlayerOutID) REFERENCES
        GamePlayer(gamePlayerID),

        CONSTRAINT GamePlayerIn_ID_FK FOREIGN KEY (gamePlayerInID) REFERENCES
GamePlayer(gamePlayerID)

);
```

Result of syntax in Oracle:

SUBSTITUTION

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
SUBSTITUTIONID	CHAR(6)	No	-	1
GAMEPLAYEROUTID	CHAR(6)	No	-	-
GAMEPLAYERINID	CHAR(6)	No	-	-
MINUTE	NUMBER(3,0)	No	-	-
REASON	VARCHAR2(30)	Yes	-	-

Download | Print

SUBSTITUTION

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205461	Check	"SUBSTITUTIONID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:30 AM	-	-
SYS_C00205462	Check	"GAMEPLAYEROUTID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:30 AM	-	-
SYS_C00205463	Check	"GAMEPLAYERINID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:30 AM	-	-
SYS_C00205464	Check	"MINUTE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:09:30 AM	-	-
SUBSTITUTION_PK	Primary	-	-	SUBSTITUTIONID	-	ENABLED	09/23/2025 09:09:30 AM	SUBSTITUTION_PK	-
GAMEPLAYEROUT_ID_FK	Foreign	-	GAMEPLAYEROUT_ID_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEPLAYER)	GAMEPLAYEROUTID	NO ACTION	ENABLED	09/23/2025 09:09:30 AM	-	-
GAMEPLAYERIN_ID_FK	Foreign	-	GAMEPLAYERIN_ID_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEPLAYER)	GAMEPLAYERINID	NO ACTION	ENABLED	09/23/2025 09:09:30 AM	-	-

AwardedCard

The SQL Syntax:

```
CREATE TABLE AwardedCard (

        awardedCardID CHAR(5) NOT NULL,

        cardDescription VARCHAR2(6) NOT NULL,

        minute NUMBER(3,0) NOT NULL,

        gamePlayerID CHAR(6) NOT NULL,

        gameRefereeID CHAR(6) NOT NULL,

        CONSTRAINT AwardedCard_PK PRIMARY KEY (awardedCardID),
```

```
        CONSTRAINT AwardedCard_GamePlayer_FK FOREIGN KEY (gamePlayerID) REFERENCES
GamePlayer(gamePlayerID),

        CONSTRAINT AwardedCard_GameReferee_FK FOREIGN KEY (gameRefereeID) REFERENCES
GameReferee(gameRefereeID)

);
```

Result of syntax in Oracle:

AWARDEDCARD

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
AWARDEDCARDID	CHAR(5)	No	-	1
CARDDESCRIPTION	VARCHAR2(6)	No	-	-
MINUTE	NUMBER(3,0)	No	-	-
GAMEPLAYERID	CHAR(6)	No	-	-
GAMEREFEREEID	CHAR(6)	No	-	-

Download | Print

AWARDEDCARD

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205468	Check	"AWARDEDCARDID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:11:51 AM	-	-
SYS_C00205469	Check	"CARDDESCRIPTION" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:11:51 AM	-	-
SYS_C00205470	Check	"MINUTE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:11:51 AM	-	-
SYS_C00205471	Check	"GAMEPLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:11:51 AM	-	-
SYS_C00205472	Check	"GAMEREFEREEID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:11:51 AM	-	-
AWARDEDCARD_PK	Primary	-	-	AWARDEDCARDID	-	ENABLED	09/23/2025 09:11:51 AM	AWARDEDCARD_PK	-
AWARDEDCARD_GAMEPLAYER_FK	Foreign	-	AWARDEDCARD_GAMEPLAYER_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEPLAYER)	GAMEPLAYERID	NO ACTION	ENABLED	09/23/2025 09:11:51 AM	-	-
AWARDEDCARD_GAMEREFEREE_FK	Foreign	-	AWARDEDCARD_GAMEREFEREE_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEREFEREE)	GAMEREFEREEID	NO ACTION	ENABLED	09/23/2025 09:11:51 AM	-	-

ScoredGoal

The SQL Syntax:

```
CREATE TABLE ScoredGoal (

        goalID CHAR(5) NOT NULL,

        gamePlayerID CHAR(6) NOT NULL,

        gameMinute NUMBER(3,0) NOT NULL,

        CONSTRAINT ScoredGoal_PK PRIMARY KEY (goalID),

        CONSTRAINT ScoredGoal_GamePlayer_FK FOREIGN KEY (gamePlayerID) REFERENCES
        GamePlayer(gamePlayerID)

);
```

Result of syntax in Oracle:

SCOREDGOAL

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
GOALID	CHAR(5)	No	-	1
GAMEPLAYERID	CHAR(6)	No	-	-
GAMEMINUTE	NUMBER(3,0)	No	-	-

Download | Print

SCOREDGOAL

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205476	Check	"GOALID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:12:26 AM	-	-
SYS_C00205477	Check	"GAMEPLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:12:26 AM	-	-
SYS_C00205478	Check	"GAMEMINUTE" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:12:26 AM	-	-
SCOREDGOAL_PK	Primary	-	-	GOALID	-	ENABLED	09/23/2025 09:12:26 AM	SCOREDGOAL_PK	-
SCOREDGOAL_GAMEPLAYER_FK	Foreign	-	SCOREDGOAL_GAMEPLAYER_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEPLAYER)	GAMEPLAYERID	NO ACTION	ENABLED	09/23/2025 09:12:26 AM	-	-

BogPlayer

The SQL Syntax:

```
CREATE TABLE BOGPlayer (

        bogPlayerID CHAR(6) NOT NULL,

        bogID CHAR(3) NOT NULL,

        gamePlayerID CHAR(6) NOT NULL,

        CONSTRAINT BogPlayer_PK PRIMARY KEY (bogPlayerID),

        CONSTRAINT BogPlayer_BOG_FK FOREIGN KEY (bogID) REFERENCES BestOnGround(bogID),

        CONSTRAINT BogPlayer_GamePlayer_FK FOREIGN KEY (gamePlayerID) REFERENCES
        GamePlayer(gamePlayerID)

);
```

Result of syntax in Oracle:

BOGPLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Add Column | Modify Column | Rename Column | Drop Column | Rename | Copy | Drop | Truncate | Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
BOGPLAYERID	CHAR(6)	No	-	1
BOGID	CHAR(3)	No	-	-
GAMEPLAYERID	CHAR(6)	No	-	-

Download | Print

BOGPLAYER

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL | REST

Create | Drop | Enable | Disable

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
SYS_C00205495	Check	"BOGPLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:41:06 AM	-	-
SYS_C00205496	Check	"BOGID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:41:06 AM	-	-
SYS_C00205497	Check	"GAMEPLAYERID" IS NOT NULL	-	-	-	ENABLED	09/23/2025 09:41:06 AM	-	-
BOGPLAYER_PK	Primary	-	-	BOGPLAYERID	-	ENABLED	09/23/2025 09:41:06 AM	BOGPLAYER_PK	-
BOGPLAYER_BOG_FK	Foreign	-	BOGPLAYER_BOG_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.BESTONGROUND)	BOGID	NO ACTION	ENABLED	09/23/2025 09:41:06 AM	-	-
BOGPLAYER_GAMEPLAYER_FK	Foreign	-	BOGPLAYER_GAMEPLAYER_FK (DBISYS5000GRP13_ISYS5000_2025_SEM2.GAMEPLAYER)	GAMEPLAYERID	NO ACTION	ENABLED	09/23/2025 09:41:06 AM	-	-

3.2 INSERT data SQL Syntax

Coach

The SQL Syntax:

INSERT ALL

```
 INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate
INSERT ALL

INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)

VALUES ('CH01','Robert','Taylor','0415789833','robert.t@email.com',TO_DATE('04/01/1968','MM/DD/YYYY'),'M',TO_DATE('01/15/2001','MM/DD/YYYY'))

INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)

VALUES ('CH02','Emily','Clarke','0413942657','emily.c@email.com',TO_DATE('09/14/1970','MM/DD/YYYY'),'F',TO_DATE('06/09/2003','MM/DD/YYYY'))

INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)

VALUES ('CH03','Anthony','Li','0413390691','anthony.li@email.com',TO_DATE('12/21/1969','MM/DD/YYYY'),'M',TO_DATE('02/20/2006','MM/DD/YYYY'))

...

INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)

VALUES ('CH07','Peter','Johnson','0412374254','peter.j@email.com',TO_DATE('01/05/1971','MM/DD/YYYY'),'M',TO_DATE('03/28/2005','MM/DD/YYYY'))

INTO Coach (coachID, firstName, lastName, phoneNum, email, dateOfBirth, gender, startDate)

VALUES ('CH08','Amanda','Wright','0411161045','amanda.w@email.com',TO_DATE('10/20/1974','MM/DD/YYYY'),'F',TO_DATE('07/30/2012','MM/DD/YYYY'))

SELECT * FROM dual;
```

Result of syntax in Oracle:

COACHID	FIRSTNAME	LASTNAME	PHONENUM	EMAIL	DATEOFBIRTH	GENDER	STARTDATE
CH01	Robert	Taylor	0415789833	robert.t@email.com	04/01/1968	M	01/15/2001
CH02	Emily	Clarke	0413942657	emily.c@email.com	09/14/1970	F	06/09/2003
CH03	Anthony	Li	0413390691	anthony.li@email.com	12/21/1969	M	02/20/2006
CH04	Sarah	Hughes	0416205432	sarah.h@email.com	07/11/1972	F	11/05/2008
CH05	Michael	Connor	0411396559	michael.oc@email.com	02/18/1967	M	09/01/2002
CH06	Lisa	Nguyen	0416751701	lisa.nguyen@email.com	06/29/1973	F	04/12/2010
CH07	Peter	Johnson	0412374254	peter.j@email.com	01/05/1971	M	03/28/2005
CH08	Amanda	Wright	0411161045	amanda.w@email.com	10/20/1974	F	07/30/2012

8 rows returned in 0.00 seconds Download

CompetitionWeek

The SQL Syntax:

```
INSERT ALL

INTO CompetitionWeek (weekID, description, startDate, endDate)

VALUES ('WK01','Opening Round',TO_DATE('05/12/2025','MM/DD/YYYY'),TO_DATE('05/18/2025','MM/DD/YYYY'))
```

```
INTO CompetitionWeek (weekID, description, startDate, endDate)

VALUES ('WK02','Rivalry Round',TO_DATE('05/19/2025','MM/DD/YYYY'),TO_DATE('05/25/2025','MM/DD/YYYY'))

INTO CompetitionWeek (weekID, description, startDate, endDate)

VALUES ('WK03','Home Advantage Round',TO_DATE('05/26/2025','MM/DD/YYYY'),TO_DATE('06/01/2025','MM/DD/YYYY'))

...

VALUES ('WK12','Penultimate Round',TO_DATE('07/28/2025','MM/DD/YYYY'),TO_DATE('08/03/2025','MM/DD/YYYY'))

INTO CompetitionWeek (weekID, description, startDate, endDate)

VALUES ('WK13','Climactic Round',TO_DATE('08/04/2025','MM/DD/YYYY'),TO_DATE('08/10/2025','MM/DD/YYYY'))

INTO CompetitionWeek (weekID, description, startDate, endDate)

VALUES ('WK14','Final Matchweek',TO_DATE('08/11/2025','MM/DD/YYYY'),TO_DATE('08/17/2025','MM/DD/YYYY'))

SELECT * FROM dual;
```

Result of syntax in Oracle:

WEEKID	DESCRIPTION	STARTDATE	ENDDATE
WK01	Opening Round	05/12/2025	05/18/2025
WK02	Rivalry Round	05/19/2025	05/25/2025
WK03	Home Advantage Round	05/26/2025	06/01/2025
WK04	Mid-Season Clash	06/02/2025	06/08/2025

....

WK11	Final Push Round	07/21/2025	07/27/2025
WK12	Penultimate Round	07/28/2025	08/03/2025
WK13	Climactic Round	08/04/2025	08/10/2025
WK14	Final Matchweek	08/11/2025	08/17/2025

14 rows returned in 0.01 seconds Download

Referee

The SQL Syntax:

```
INSERT ALL

INTO Referee (refereeID, firstName, lastName, dateOfBirth, phoneNum, startDate, gender, refLevel)

VALUES
('RF01','Sarah','Wilson',TO_DATE('06/12/1984','MM/DD/YYYY'),'0404000001',TO_DATE('03/01/2010','MM/DD/YYYY'),'F',1)

INTO Referee (refereeID, firstName, lastName, dateOfBirth, phoneNum, startDate, gender, refLevel)
```

```
VALUES
('RF02','Michael','Jones',TO_DATE('11/22/1981','MM/DD/YYYY'),'0404000002',TO_DATE('05/10/2008','MM/DD/YYYY'),'M',1)

...

INTO Referee (refereeID, firstName, lastName, dateOfBirth, phoneNum, startDate, gender, refLevel)

VALUES
('RF11','Grace','Martin',TO_DATE('11/06/1994','MM/DD/YYYY'),'0404000011',TO_DATE('04/10/2019','MM/DD/YYYY'),'F',4)

INTO Referee (refereeID, firstName, lastName, dateOfBirth, phoneNum, startDate, gender, refLevel)

VALUES
('RF12','Thomas','Baker',TO_DATE('02/23/1979','MM/DD/YYYY'),'0404000012',TO_DATE('09/18/2006','MM/DD/YYYY'),'M',3)

SELECT * FROM dual;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

REFEREEID	FIRSTNAME	LASTNAME	DATEOFBIRTH	PHONENUM	STARTDATE	GENDER	REFLEVEL
RF01	Sarah	Wilson	06/12/1984	0404000001	03/01/2010	F	1
RF02	Michael	Jones	11/22/1981	0404000002	05/10/2008	M	1
RF03	Linda	Tran	03/15/1988	0404000003	07/20/2015	F	2
RF04	James	Patel	09/30/1983	0404000004	02/14/2011	M	1

....

RF10	William	Taylor	10/17/1988	0404000010	03/02/2014	M	2
RF11	Grace	Martin	11/06/1994	0404000011	04/10/2019	F	4
RF12	Thomas	Baker	02/23/1979	0404000012	09/18/2006	M	3

12 rows returned in 0.00 seconds Download

StartPosition

The SQL Syntax:

```
INSERT ALL

INTO StartPosition (positionID, description) VALUES ('SP01','Goalkeeper')

INTO StartPosition (positionID, description) VALUES ('SP02','Right Back')

INTO StartPosition (positionID, description) VALUES ('SP03','Left Back')

INTO StartPosition (positionID, description) VALUES ('SP10','Left Wing')

INTO StartPosition (positionID, description) VALUES ('SP11','Center Forward')

INTO StartPosition (positionID, description) VALUES ('SP12','Substitute')

SELECT * FROM dual;
```

Result of syntax in Oracle:

POSITIONID	DESCRIPTION
SP01	Goalkeeper
SP02	Right Back
SP03	Left Back
SP04	Center Back
....	
SP09	Right Wing
SP10	Left Wing
SP11	Center Forward
SP12	Substitute

12 rows returned in 0.00 seconds Download

BestOnGround

The SQL Syntax:

```
INSERT ALL

INTO BestOnGround (bestOnGroundID, rank, points) VALUES ('BG1', 1, 3)

INTO BestOnGround (bestOnGroundID, rank, points) VALUES ('BG2', 2, 2)

INTO BestOnGround (bestOnGroundID, rank, points) VALUES ('BG3', 3, 1)

SELECT * FROM dual;
```

Result of syntax in Oracle:

BOGID	RANK	POINTS
BG1	1	3
BG2	2	2
BG3	3	1

3 rows returned in 0.00 seconds Download

Result

The SQL Syntax:

```
I NSERT ALL

INTO Result (resultID, description, points) VALUES ('RS01','Away Win',6)

INTO Result (resultID, description, points) VALUES ('RS02','Home Win',3)

INTO Result (resultID, description, points) VALUES ('RS03','Away Draw',2)

INTO Result (resultID, description, points) VALUES ('RS04','Home Draw',1)

INTO Result (resultID, description, points) VALUES ('RS05','Away Loss',0)

INTO Result (resultID, description, points) VALUES ('RS06','Home Loss',0)
```

```
SELECT * FROM dual;
```

Result of syntax in Oracle:

RESULTID	DESCRIPTION	POINTS
RS01	Away Win	6
RS02	Home Win	3
RS03	Away Draw	2
RS04	Home Draw	1
RS05	Away Loss	0
RS06	Home Loss	0

6 rows returned in 0.01 seconds Download

Club

The SQL Syntax:

```
INSERT ALL

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL01','Eagle Bay Eagles',1985,'Busselton Motors',120,'CH01')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL02','Dunsborough Dingos',1978,'SurfCo Insurance',115,'CH02')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL03','Broadwater Bandicoots',1990,'Busselton Fresh',130,'CH03')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL04','Mary-Brook Magpies',1982,'Brook Wines',110,'CH04')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL05','Reinscourt Rabbits',1988,'West Coast Bank',125,'CH05')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL06','Wonnerup Wombats',1975,'Wonn Chemicals',118,'CH06')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL07','New Caves Rd Numbats',1993,'Cave Tours WA',112,'CH07')

INTO Club (clubID, clubName, yearEstablished, majorSponsor, numOfMembers, coachID)

VALUES ('CL08','Hanes Hawkes',1980,'Hanes Builders',128,'CH08')

SELECT * FROM DUAL;
```

Result of syntax in Oracle:

CLUBID	CLUBNAME	YEARESTABLISHED	MAJORSPONSOR	NUMOFMEMBERS	COACHID
CL01	Eagle Bay Eagles	1985	Busselton Motors	120	CH01
CL02	Dunsborough Dingos	1978	SurfCo Insurance	115	CH02
CL03	Broadwater Bandicoots	1990	Busselton Fresh	130	CH03
CL04	Mary-Brook Magpies	1982	Brook Wines	110	CH04
CL05	Reinscourt Rabbits	1988	West Coast Bank	125	CH05
CL06	Wonnerup Wombats	1975	Wonn Chemicals	118	CH06
CL07	New Caves Rd Numbats	1993	Cave Tours WA	112	CH07
CL08	Hanes Hawkes	1980	Hanes Builders	128	CH08

8 rows returned in 0.00 seconds Download

Player

The SQL Syntax:

```
I INSERT ALL

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL101','Liam','Taylor','0495888726','liam.taylor1@email.com','80',TO_DATE('10-06-98','DD-MM-RR'),175,75,'CL01')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL102','Martin','Carter','0422519875','martin.carter2@email.com','60',TO_DATE('04-03-00','DD-MM-RR'),187,79,'CL01')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL103','Stuart','Martin','0416138399','stuart.martin3@email.com','98',TO_DATE('22-12-01','DD-MM-RR'),174,76,'CL01')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL104','Mark','Wright','0427054234','mark.wright4@email.com','83',TO_DATE('20-04-92','DD-MM-RR'),177,79,'CL01')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

....

VALUES ('PL809','George','Evans','0458920186','george.evans93@email.com','34',TO_DATE('20-10-96','DD-MM-RR'),187,82,'CL08')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL810','Chris','Brooks','0425974901','chris.brooks94@email.com','3',TO_DATE('01-07-99','DD-MM-RR'),177,69,'CL08')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)
```

```
VALUES ('PL811','Daniel','Carter','0416421248','daniel.carter95@email.com','72',TO_DATE('26-08-97','DD-MM-RR'),190,80,'CL08')

INTO Player (playerID, firstName, lastName, phoneNum, email, jerseyNumber, dateOfBirth, height, weight, clubID)

VALUES ('PL812','Peter','Foster','0420056507','peter.foster96@email.com','16',TO_DATE('24-10-94','DD-MM-RR'),180,79,'CL08')

SELECT * FROM DUAL;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

PLAYERID	FIRSTNAME	LASTNAME	PHONENUM	EMAIL	JERSEYNUMBER	DATEOFBIRTH	HEIGHT	WEIGHT	CLUBID
PL101	Liam	Taylor	0495888726	liam.taylor1@email.com	80	06/10/1998	175	75	CL01
PL102	Martin	Carter	0422519875	martin.carter2@email.com	60	03/04/2000	187	79	CL01
PL103	Stuart	Martin	0416138399	stuart.martin3@email.com	98	12/22/2001	174	76	CL01
PL104	Mark	Wright	0427054234	mark.wright4@email.com	83	04/20/1992	177	79	CL01
PL105	Alan	Hughes	0465164748	alan.hughes5@email.com	49	05/15/1999	173	62	CL01
PL106	Martin	Wilson	0414826896	martin.wilson6@email.com	33	01/17/2004	184	80	CL01

...

PL809	George	Evans	0458920186	george.evans93@email.com	34	10/20/1996	187	82	CL08
PL810	Chris	Brooks	0425974901	chris.brooks94@email.com	3	07/01/1999	177	69	CL08
PL811	Daniel	Carter	0416421248	daniel.carter95@email.com	72	08/26/1997	190	80	CL08
PL812	Peter	Foster	0420056507	peter.foster96@email.com	16	10/24/1994	180	79	CL08

96 rows returned in 0.01 seconds Download

Ground

The SQL Syntax:

```
INSERT ALL

INTO Ground (groundID, name, address, capacity, clubID)

VALUES ('GR01','Eagle Bay Centre','Eagle Bay Rd, Busselton WA',2000,'CL01')

INTO Ground (groundID, name, address, capacity, clubID)

VALUES ('GR02','Quedjinup Reserve','Naturaliste Terrace, WA',1500,'CL02')

INTO Ground (groundID, name, address, capacity, clubID)

VALUES ('GR03','Broadwater Bay Oval','Broadwater WA',2200,'CL03')

INTO Ground (groundID, name, address, capacity, clubID)

VALUES ('GR06','Wonnerup Sports Ground','Wonnerup WA',1700,'CL06')

INTO Ground (groundID, name, address, capacity, clubID)

VALUES ('GR07','Caves Road Reserve','Caves Rd, Busselton WA',1600,'CL07')

INTO Ground (groundID, name, address, capacity, clubID)
```

```
VALUES ('GR08','Hanes Sports Ground','Hanes WA',1900,'CL08')

SELECT * FROM dual;
```

Result of syntax in Oracle:

GROUNDID	NAME	ADDRESS	CAPACITY	CLUBID
GR01	Eagle Bay Centre	Eagle Bay Rd, Busselton WA	2000	CL01
GR02	Quedjinup Reserve	Naturaliste Terrace, WA	1500	CL02
GR03	Broadwater Bay Oval	Broadwater WA	2200	CL03
GR04	Kealy Reserve	Kealy WA	1800	CL04
GR05	West Busselton Oval	West Busselton WA	2100	CL05
GR06	Wonnerup Sports Ground	Wonnerup WA	1700	CL06
GR07	Caves Road Reserve	Caves Rd, Busselton WA	1600	CL07
GR08	Hanes Sports Ground	Hanes WA	1900	CL08

8 rows returned in 0.00 seconds Download

Game

The SQL Syntax:

```
INSERT ALL

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM01','CL01 v
CL02',TO_DATE('05/12/2025','MM/DD/YYYY'),TO_DATE('08:00','HH24:MI'),TO_DATE('09:30','HH24:MI'),'WK01','G
R01')

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM02','CL03 v
CL04',TO_DATE('05/14/2025','MM/DD/YYYY'),TO_DATE('14:00','HH24:MI'),TO_DATE('15:30','HH24:MI'),'WK01','G
R03')

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM03','CL05 v
CL06',TO_DATE('05/16/2025','MM/DD/YYYY'),TO_DATE('08:00','HH24:MI'),TO_DATE('09:30','HH24:MI'),'WK01','G
R05')

....

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM54','CL03 v
CL02',TO_DATE('08/13/2025','MM/DD/YYYY'),TO_DATE('14:00','HH24:MI'),TO_DATE('15:30','HH24:MI'),'WK14','G
R03')

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM55','CL07 v
CL04',TO_DATE('08/15/2025','MM/DD/YYYY'),TO_DATE('08:00','HH24:MI'),TO_DATE('09:30','HH24:MI'),'WK14','G
R07')

INTO Game (GAMEID, DESCRIPTION, GAMEDATE, STARTTIME, ENDTIME, WEEKID, GROUNDID) VALUES
('GM56','CL06 v
CL05',TO_DATE('08/17/2025','MM/DD/YYYY'),TO_DATE('14:00','HH24:MI'),TO_DATE('15:30','HH24:MI'),'WK14','G
R06')

SELECT * FROM dual;
```

** The game table was later updated to insert data for numOfSpectators **

```
BEGIN
UPDATE Game SET numOfSpectators = 1850 WHERE gameID = 'GM01';

UPDATE Game SET numOfSpectators = 1200 WHERE gameID = 'GM02';

UPDATE Game SET numOfSpectators = 1650 WHERE gameID = 'GM03';

UPDATE Game SET numOfSpectators = 950 WHERE gameID = 'GM04';

UPDATE Game SET numOfSpectators = 1920 WHERE gameID = 'GM05';

...

UPDATE Game SET numOfSpectators = 1280 WHERE gameID = 'GM52';

UPDATE Game SET numOfSpectators = 1640 WHERE gameID = 'GM53';

UPDATE Game SET numOfSpectators = 1980 WHERE gameID = 'GM54';

UPDATE Game SET numOfSpectators = 1020 WHERE gameID = 'GM55';

UPDATE Game SET numOfSpectators = 1420 WHERE gameID = 'GM56';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

GAMEID	DESCRIPTION	GAMEDATE	STARTTIME	ENDTIME	WEEKID	GROUNDID
GM01	CL01 v CL02	05/12/2025	09/01/2025	09/01/2025	WK01	GR01
GM02	CL03 v CL04	05/14/2025	09/01/2025	09/01/2025	WK01	GR03
GM03	CL05 v CL06	05/16/2025	09/01/2025	09/01/2025	WK01	GR05
GM04	CL07 v CL08	05/18/2025	09/01/2025	09/01/2025	WK01	GR07
GM05	CL01 v CL03	05/19/2025	09/01/2025	09/01/2025	WK02	GR01
...						
GM54	CL03 v CL02	08/13/2025	09/01/2025	09/01/2025	WK14	GR03
GM55	CL07 v CL04	08/15/2025	09/01/2025	09/01/2025	WK14	GR07
GM56	CL06 v CL05	08/17/2025	09/01/2025	09/01/2025	WK14	GR06

56 rows returned in 0.00 seconds Download

ClubGame

The SQL Syntax:

```
INSERT ALL
```

```
INTO ClubGame (clubGameID, gameID, clubID, goalScored, resultID)

VALUES ('CG001','GM01','CL01',4,'RS02')

INTO ClubGame VALUES ('CG002','GM01','CL02',1,'RS05')

INTO ClubGame VALUES ('CG003','GM02','CL03',1,'RS04')

INTO ClubGame VALUES ('CG004','GM02','CL04',1,'RS03')

....

INTO ClubGame VALUES ('CG110','GM55','CL03',1,'RS01')

INTO ClubGame VALUES ('CG111','GM56','CL05',1,'RS04')

INTO ClubGame VALUES ('CG112','GM56','CL04',1,'RS03')

SELECT * FROM DUAL;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

CLUBGAMEID	GAMEID	CLUBID	GOALSCORED	RESULTID
CG001	GM01	CL01	4	RS02
CG002	GM01	CL02	1	RS05
CG003	GM02	CL03	1	RS04
CG004	GM02	CL04	1	RS03
CG005	GM03	CL05	1	RS06

...

CG110	GM55	CL03	1	RS01
CG111	GM56	CL05	1	RS04
CG112	GM56	CL04	1	RS03

112 rows returned in 0.01 seconds Download

GameReferee

The SQL Syntax:

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

GAMEREFEREEID	REFROLE	GAMEID	REFEREEID
GRF001	Main	GM01	RF01
GRF002	Assistant1	GM01	RF02
GRF003	Assistant2	GM01	RF03
GRF004	Main	GM02	RF04
GRF005	Assistant1	GM02	RF05

...

GRF166	Main	GM56	RF04
GRF167	Assistant1	GM56	RF05
GRF168	Assistant2	GM56	RF08

168 rows returned in 0.01 seconds Download

GamePlayer

The SQL Syntax:

```
INSERT ALL

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP0001','GM01','PL101','SP01',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP0002','GM01','PL102','SP02',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP0003','GM01','PL103','SP03',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP0004','GM01','PL104','SP04',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP0005','GM01','PL105','SP05',90)

...

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP1340','GM56','PL408','SP08',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP1341','GM56','PL409','SP09',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP1342','GM56','PL410','SP10',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP1343','GM56','PL411','SP11',90)

INTO GamePlayer (gamePlayerID, gameID, playerID, positionID, gameMinute)

VALUES ('GP1344','GM56','PL412','SP12',9)

SELECT * FROM dual;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

GAMEPLAYERID	GAMEID	PLAYERID	POSITIONID	GAMEMINUTE
GP0001	GM01	PL101	SP01	90
GP0002	GM01	PL102	SP02	90
GP0003	GM01	PL103	SP03	90
GP0004	GM01	PL104	SP04	90
GP0005	GM01	PL105	SP05	90
...				
GP1342	GM56	PL410	SP10	90
GP1343	GM56	PL411	SP11	90
GP1344	GM56	PL412	SP12	0

1344 rows returned in 0.05 seconds Download

Substitution

The SQL Syntax:

```
INSERT ALL

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB001', 'GP0022', 'GP0024', 69, 'Fatigue')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB002', 'GP0045', 'GP0048', 26, 'Performance issue')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB003', 'GP0053', 'GP0060', 24, 'Injury concern')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB004', 'GP0087', 'GP0096', 80, 'Fatigue')

....

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB053', 'GP1254', 'GP1260', 70, 'Injury concern')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB054', 'GP1279', 'GP1284', 24, 'Tactical change')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB055', 'GP1299', 'GP1308', 59, 'Strategic substitution')

INTO Substitution (substitutionID, gamePlayerOutID, gamePlayerInID, minute, reason) VALUES ('SUB056', 'GP1330', 'GP1332', 9, 'Fatigue')

SELECT * FROM dual;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

SUBSTITUTIONID	GAMEPLAYEROUTID	GAMEPLAYERINID	MINUTE	REASON
SUB001	GP0022	GP0024	69	Fatigue
SUB002	GP0045	GP0048	26	Performance issue
SUB003	GP0053	GP0060	24	Injury concern
SUB004	GP0087	GP0096	80	Fatigue

...

SUB054	GP1279	GP1284	24	Tactical change
SUB055	GP1299	GP1308	59	Strategic substitution
SUB056	GP1330	GP1332	9	Fatigue

56 rows returned in 0.00 seconds Download

AwardedCard

The SQL Syntax:

```
INSERT ALL

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC001', 'Yellow', 15, 'GP0010', 'GRF001')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC002', 'Yellow', 22, 'GP0009', 'GRF001')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC003', 'Yellow', 25, 'GP0009', 'GRF001')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC004', 'Yellow', 28, 'GP0025', 'GRF004')

....

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC012', 'Yellow', 76, 'GP0990', 'GRF124')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC013', 'Yellow', 42, 'GP0995', 'GRF124')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('AC014', 'Yellow', 74, 'GP0995', 'GRF124')

INTO AwardedCard (awardedCardID, cardDescription, minute, gamePlayerID, gameRefereeID) VALUES
('ACO15', 'Red', 50, 'GP0220', 'GRF028')

SELECT * FROM dual;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

AWARDEDCARDID	CARDDESCRIPTION	MINUTE	GAMEPLAYERID	GAMEREFEREEID
AC001	Yellow	15	GP0010	GRF001
AC002	Yellow	22	GP0009	GRF001
AC003	Yellow	25	GP0009	GRF001
AC004	Yellow	28	GP0025	GRF004

...

AC012	Yellow	76	GP0990	GRF124
AC013	Yellow	42	GP0995	GRF124
AC014	Yellow	74	GP0995	GRF124
ACO15	Red	50	GP0220	GRF028

13 rows returned in 0.00 seconds Download

ScoredGoal

The SQL Syntax:

```
INSERT ALL

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL001', 'GP0010', 44)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL002', 'GP0010', 66)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL003', 'GP0010', 78)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL004', 'GP0022', 40)

....

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL150', 'GP1283', 89)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL151', 'GP1319', 65)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL152', 'GP1342', 77)

INTO ScoredGoal (goalID, gamePlayerID, gameMinute) VALUES ('GL153', 'GP1328', 83)

SELECT * FROM dual;
```

Result of syntax in Oracle:

Results | Explain | Describe | Saved SQL | History

GOALID	GAMEPLAYERID	GAMEMINUTE
GL001	GP0010	44
GL002	GP0010	66
GL003	GP0010	78
GL004	GP0022	40
GL005	GP0011	89

...

GL150	GP1283	89
GL151	GP1319	65
GL152	GP1342	77
GL153	GP1328	83

153 rows returned in 0.00 seconds Download

BogPlayer

The SQL Syntax:

```
INSERT ALL

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0001', 'BG1', 'GP0010')

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0002', 'BG2', 'GP0022')

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0003', 'BG3', 'GP0011')

....

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0166', 'BG1', 'GP1342')

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0167', 'BG2', 'GP1328')

INTO BogPlayer (bogPlayerID, bogID, gamePlayerID) VALUES ('BP0168', 'BG3', 'GP1331')

SELECT * FROM dual;
```

Result of syntax in Oracle:

BOGPLAYERID	BOGID	GAMEPLAYERID
BP0001	BG1	GP0010
BP0002	BG2	GP0022
BP0003	BG3	GP0011
BP0004	BG1	GP0046

...

BP0165	BG3	GP1316
BP0166	BG1	GP1342
BP0167	BG2	GP1328
BP0168	BG3	GP1331

168 rows returned in 0.00 seconds Download

4. SQL Statements/Exercises

Write SQL scripts to test the functionality and effectiveness of the database you have created. You can include comments with your scripts, informing the purpose of the script, why you wrote it the way you did, who contributed and issues you encountered or problems you had to solve. These notes will be especially important if you can't get a script to work properly, as it will provide the only evidence of your thinking to the marker of your work.

Each Script should be presented in text – showing the SQL* used, underneath the appropriate heading (so it is obvious which question you are answering) followed by a screen-grab – using a

similar approach to the example demonstrated in figure 3 – followed by a description/explanation of the SQL used.

* NOTE: The SQL coding will be copied and tested by the marker. The coding must firstly actually work, and secondly answer the question properly, to gain marks.

4.1 Season Outcomes

4.1a Use the data in the various tables to find which two clubs would take part in a post-season game between the clubs that came first and second after 14 weeks of competition. You should include aggregated information such as total points scored, away and home results, total goals scored and the like.

Syntax for query 4.1a

```
SELECT

cl.clubName AS "Club Name",

crs.total_result_points AS "Result Points",

bogp.total_bog_points AS "BOG Points",

tgs.goals_scored_count AS "Goals Scored",

hws.home_win_count AS "Home Wins",

aws.away_win_count AS "Away Wins",

crs.total_result_points + bogp.total_bog_points AS "Total Points (Result + BOG)"

FROM Club cl

JOIN (

SELECT cg.clubID, SUM(rs.points) AS total_result_points

FROM ClubGame cg

JOIN Result rs ON rs.resultID = cg.resultID

GROUP BY cg.clubID

) crs ON crs.clubID = cl.clubID

JOIN (

SELECT pl.clubID, SUM(bog.points) AS total_bog_points

FROM BogPlayer bp

JOIN BestOnGround bog ON bog.bogID = bp.bogID

JOIN GamePlayer gp ON gp.gamePlayerID = bp.gamePlayerID

JOIN Player pl ON pl.playerID = gp.playerID

GROUP BY pl.clubID

) bogp ON bogp.clubID = cl.clubID

JOIN (

SELECT cg.clubID, SUM(cg.goalScored) AS goals_scored_count
```

```
FROM ClubGame cg

GROUP BY cg.clubID

) tgs ON tgs.clubID = cl.clubID

JOIN (

SELECT cg.clubID, COUNT(*) AS home_win_count

FROM ClubGame cg

WHERE cg.resultID = 'RS02'

GROUP BY cg.clubID

) hws ON hws.clubID = cl.clubID

JOIN (

SELECT cg.clubID, COUNT(*) AS away_win_count

FROM ClubGame cg

WHERE cg.resultID = 'RS01'

GROUP BY cg.clubID

) aws ON aws.clubID = cl.clubID

ORDER BY "Result Points" DESC;
```

Screen-grab/Result of query 4.1a

Results | Explain | Describe | Saved SQL | History

Club Name	Result Points	BOG Points	Goals Scored	Home Wins	Away Wins	Total Points (Result + BOG)
Wonnerup Wombats	38	45	20	3	4	83
Eagle Bay Eagles	37	46	22	5	3	83
Hanes Hawkes	36	43	17	2	4	79
Dunsborough Dingos	32	44	20	2	4	76
Broadwater Bandicoots	29	41	17	1	3	70
Mary-Brook Magpies	26	40	19	1	3	66
New Caves Rd Numbats	23	39	18	3	2	62
Reinscourt Rabbits	22	38	20	1	2	60

8 rows returned in 0.02 seconds Download

Explanation of query 4.1a

- The SELECT query has one main table which is Club (cl).
- The SQL query retrieves the clubName from the Club (cl) table.
- Subqueries are used to calculate additional information for each club:

  From ClubGame (cg) and Result (rs) tables to SUM the result points as Result Points (alias crs).
  From BogPlayer (bp), BestOnGround (bog), GamePlayer (gp) and Player (pl) tables to SUM the Best-on-Ground points as BOG Points (alias bogp).

  From ClubGame (cg) table to SUM the goals scored as Goals Scored (alias tgs).

From ClubGame (cg) table, filtered where resultID = 'RS02', to COUNT the number of home wins as Home Wins (alias hws).

From ClubGame (cg) table, filtered where resultID = 'RS01', to COUNT the number of away wins as Away Wins (alias aws).

The query also calculates a Total Points (Result + BOG) column by adding together the result points and the BOG points, giving an overall score that can be used to compare clubs.

- The query uses aliases like cl, cg, rs, bp, bog, gp, and pl to make the SQL shorter and easier to read, instead of repeating full table names every time.
- Each subquery applies GROUP BY clubID so that all the rows for one club are grouped together, giving a single total for things like points or goals.
- The results from these subqueries are then joined back to the Club (cl) table using clubID, so that each club's name appears alongside its calculated totals.
- The final step uses ORDER BY "Result Points" DESC to sort the output, showing the clubs with the highest points at the top and making it clear which clubs are leading.

4.1b Produce a Competition ladder for the end of the home-and-away season of games that lists the clubs in order of where they finished (from 1st to 8th), how many competition points they earned, how many home games and away games they won and/or lost, and how many total number of goals they scored for the whole season

Syntax for query 4.1b

```
SELECT cl.clubID, cl.clubName,


(SELECT SUM(rs.points)

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID

WHERE cg.clubID = cl.clubID) AS "Total Competition Point",


(SELECT SUM(cg.goalScored)

FROM ClubGame cg

WHERE cg.clubID = cl.clubID) AS "Total Goal Score",


(SELECT COUNT(*)

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID

JOIN Game gm ON cg.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID = cl.clubID AND rs.description = 'Home Win') AS "Home Win",


(SELECT COUNT(*)

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID
```

```
JOIN Game gm ON cg.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID = cl.clubID AND rs.description = 'Home Loss') AS "Home Loss",

(SELECT COUNT(*)

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID

JOIN Game gm ON cg.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID = cl.clubID AND rs.description = 'Away Win') AS "Away Win",

(SELECT COUNT(*)

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID

JOIN Game gm ON cg.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID = cl.clubID AND rs.description = 'Away Loss') AS "Away Loss"

FROM Club cl

ORDER BY "Total Competition Point" DESC, "Total Goal Score" DESC;
```

Screen-grab/Result of query 4.1b

Results | Explain | Describe | Saved SQL | History

CLUBID	CLUBNAME	Total Competition Point	Total Goal Score	Home Win	Home Loss	Away Win	Away Loss
CL06	Wonnerup Wombats	38	20	3	3	4	1
CL01	Eagle Bay Eagles	37	22	5	2	3	2
CL08	Hanes Hawkes	36	17	2	1	4	2
CL02	Dunsborough Dingos	32	20	2	3	4	3
CL03	Broadwater Bandicoots	29	17	1	4	3	1
CL04	Mary-Brook Magpies	26	19	1	5	3	2
CL07	New Caves Rd Numbats	23	18	3	4	2	4
CL05	Reinscourt Rabbits	22	20	1	3	2	3

8 rows returned in 0.01 seconds Download

Explanation of query 4.1b

- The SELECT query had one main table which is Club (cl).

- The SQL query retrieves the clubID and clubName from the Club (cl) table.

- Subqueries were used to calculate additional information for each club.

From ClubGame(cg) and Result(rs) tables to SUM the competition points AS Total Competition Point.

From ClubGame(cg) to SUM the goals scored as Total Goal Score.

From ClubGame (cg), Result(rs), Game(gm) and Ground(grn) tables to count the number of Home win, Home loss, Away win and Away loss, matching for each club.

- Using aliases cl,cg,rs,gm,grn for the Club, ClubGame, Result, Game and Ground tables respectively. These make the query shorter and easier to read instead of full writing.

-WHERE condition inside each subquery tells that cg.clubID must match the current c.clubID. Thus, only the games related to each club are included.

-ORDER BY condition sorts the results by Total Competition Point in descending order. If the points are the same, then by Total Goal Score in descending order.

4.2 Seasonal Goals & Scorers

4.2a. Provide a list of the season's goals, who scored them and in which game. List the goals in chronological, then game order.

Syntax for query 4.2a

```
SELECT

gm.gameID AS "Game Number",

gm.description AS "Fixture",

cl.clubName AS "Club Scored",

pl.firstName || ' ' || pl.lastName AS "Name of the Scorer",

sg.gameMinute AS "Minute Goal Scored",

grn.name AS "Name of Ground",

gm.gameDate AS "Date"

FROM ScoredGoal sg

JOIN GamePlayer gp ON sg.gamePlayerID = gp.gamePlayerID

JOIN Player pl ON gp.playerID = pl.playerID

JOIN ClubGame cg ON gp.gameID = cg.gameID AND cg.clubID = pl.clubID

JOIN Club cl ON cg.clubID = cl.clubID

JOIN Game gm ON gp.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

ORDER BY gm.gameDate, gm.gameID, sg.gameMinute;
```

Screen-grab/Result of query 4.2a

Results | Explain | Describe | Saved SQL | History

Game Number	Fixture	Club Scored	Name of the Scorer	Minute Goal Scored	Name of Ground	Date
GM01	CL01 v CL02	Dunsborough Dingos	Steve Connor	40	Eagle Bay Centre	05/12/2025
GM01	CL01 v CL02	Eagle Bay Eagles	Glen Evans	44	Eagle Bay Centre	05/12/2025
GM01	CL01 v CL02	Eagle Bay Eagles	Glen Evans	66	Eagle Bay Centre	05/12/2025
GM01	CL01 v CL02	Eagle Bay Eagles	Glen Evans	78	Eagle Bay Centre	05/12/2025
GM01	CL01 v CL02	Eagle Bay Eagles	Kevin Murray	89	Eagle Bay Centre	05/12/2025
GM02	CL03 v CL04	Mary-Brook Magpies	Martin Tan	39	Broadwater Bay Oval	05/14/2025
GM02	CL03 v CL04	Broadwater Bandicoots	Frank Brooks	76	Broadwater Bay Oval	05/14/2025
GM03	CL05 v CL06	Wonnerup Wombats	Frank Hughes	50	West Busselton Oval	05/16/2025
GM03	CL05 v CL06	Wonnerup Wombats	Frank Hughes	76	West Busselton Oval	05/16/2025

...

GM52	CL06 v CL04	Wonnerup Wombats	Frank Hughes	30	Wonnerup Sports Ground	08/10/2025
GM52	CL06 v CL04	Mary-Brook Magpies	Peter Connor	74	Wonnerup Sports Ground	08/10/2025
GM52	CL06 v CL04	Mary-Brook Magpies	Frank Brooks	79	Wonnerup Sports Ground	08/10/2025
GM52	CL06 v CL04	Wonnerup Wombats	David Brooks	84	Wonnerup Sports Ground	08/10/2025
GM52	CL06 v CL04	Wonnerup Wombats	Victor Johnson	89	Wonnerup Sports Ground	08/10/2025
GM54	CL03 v CL02	New Caves Rd Numbats	Glen Smith	41	Broadwater Bay Oval	08/13/2025
GM54	CL03 v CL02	New Caves Rd Numbats	Alex Martin	89	Broadwater Bay Oval	08/13/2025
GM55	CL07 v CL04	Broadwater Bandicoots	Peter Carter	65	Caves Road Reserve	08/15/2025
GM56	CL06 v CL05	Mary-Brook Magpies	Martin Tan	77	Wonnerup Sports Ground	08/17/2025
GM56	CL06 v CL05	Reinscourt Rabbits	Paul Brown	83	Wonnerup Sports Ground	08/17/2025

153 rows returned in 0.02 seconds Download

Explanation of query 4.2a

- The SELECT query has one main table which is Game (gm).
- The query shows details like the game number, fixture name, date, and ground name from the Game (gm) and Ground (grn) tables.
- Other tables are joined to add more information:

  ScoredGoal (sg) and GamePlayer (gp) are used to find which player scored the goal.

  Player (pl) is used to get the scorer's first and last name.

  ClubGame (cg) and Club (cl) are used to find which club the scorer belongs to.

  Aliases (gm, sg, gp, pl, cg, cl, grn) are used to make the query shorter and easier to read.
- The query also combines the player's first name and last name into one column so the full name is shown.
- The results are sorted by game date, game number, and the minute of the goal, so the goals are listed in the order they happened.

4.2b. Write a query that counts the number of gaols scored at each of the 8 grounds during the season. Order the list you produce from most – to – least goals scored at the ground.

Syntax for query 4.2b

```
SELECT

grn.name AS "Ground Name",
```

```
COUNT(sg.goalID) AS "Total Goals",

ROUND(COUNT(sg.goalID) / COUNT(DISTINCT gm.gameID), 0) AS "Average Goals per Game"

FROM Ground grn

JOIN Game gm ON grn.groundID = gm.groundID

JOIN GamePlayer gp ON gm.gameID = gp.gameID

LEFT JOIN ScoredGoal sg ON gp.gamePlayerID = sg.gamePlayerID

GROUP BY grn.name;
```

Screen-grab/Result of query 4.2b

Ground Name	Total Goals	Average Goals per Game
Wonnerup Sports Ground	17	2
Kealy Reserve	22	3
West Busselton Oval	19	3
Broadwater Bay Oval	16	2
Caves Road Reserve	18	3
Hanes Sports Ground	14	2
Eagle Bay Centre	25	4
Quedjinup Reserve	22	3

8 rows returned in 0.01 seconds Download

Explanation of query 4.2b

- The main table in the query is Ground (grn), which gives us the ground name.
- The query counts how many goals (sg.goalID) were scored at each ground, shown as Total Goals.
- We also calculates the average number of goals per game at each ground by dividing:
  the total number of goals scored (COUNT of sg.goalID) by the number of different games played there (COUNT of DISTINCT gm.gameID).
- DISTINCT is used here because the same game could appear many times in the data (one row for each player). Using DISTINCT makes sure each game is only counted once.
- The JOINs connect the tables together:
  Ground (grn) is joined to Game (gm) to find all games played at that ground.

  Game (gm) is joined to GamePlayer (gp) to list the players in those games.

  ScoredGoal (sg) is joined to GamePlayer (gp) to count which players actually scored goals.
- A LEFT JOIN is used so games with no goals are still included in the average.
- The GROUP BY grn.name ensures the totals and averages are calculated separately for each ground.
-

4.2c. Provide a list of the players who scored goals during the season, including the total number of goals each player scored. List this in descending order (i.e., number of goals scored) from highest number

Syntax for query 4.2c

```
SELECT

pl.firstname || ' ' || pl.lastname AS "Name",
```

```
COUNT(sg.gamePlayerID) AS "Goals Scored",

cl.clubname AS "Club Name"

FROM ScoredGoal sg

JOIN GamePlayer gp ON gp.gamePlayerID = sg.gamePlayerID

JOIN Player pl ON pl.playerID = gp.playerID

JOIN Club cl ON pl.clubID = cl.clubID

GROUP BY pl.firstname, pl.lastname, cl.clubname

ORDER BY COUNT(sg.gamePlayerID) DESC;
```

Screen-grab/Result of query 4.2c

Results Explain Describe Saved SQL History

Name	Goals Scored	Club Name
Glen Evans	9	Eagle Bay Eagles
David Martin	7	Dunsborough Dingos
Frank Brooks	6	Broadwater Bandicoots
Adam Tan	6	Dunsborough Dingos
Alan Smith	6	Reinscourt Rabbits
George Reid	6	Broadwater Bandicoots
Frank Hughes	6	Wonnerup Wombats
Paul Brown	6	Reinscourt Rabbits
Kevin Murray	6	Eagle Bay Eagles
Martin Tan	6	Mary-Brook Magpies
Alex Martin	6	New Caves Rd Numbats
Martin Connor	6	Hanes Hawkes
Glen Smith	5	New Caves Rd Numbats

...

Liam Foster	3	Mary-Brook Magpies
Jason Murray	2	Dunsborough Dingos
Kevin Hughes	2	Eagle Bay Eagles
Stuart Carter	2	New Caves Rd Numbats
Daniel Carter	2	Hanes Hawkes
Michael Wright	1	Broadwater Bandicoots
Daniel Wright	1	Dunsborough Dingos
Adam Hughes	1	Mary-Brook Magpies
Martin Wilson	1	Eagle Bay Eagles

36 rows returned in 0.00 seconds Download

Explanation of query 4.2c

- The SELECT query has one main table which is Player (pl).
- The query shows details like the player's full name and last name together, the number of goals scored, and the club name from the Club table.
- Other tables are joined to add more information:
  - ScoredGoal (sg) and GamePlayer (gp) are used to link each goal to the player who scored it.
  - Player (pl) is used to display the player's first and last name.
  - Club (cl) is used to show which club the player belongs to.
- Aliases (sg, gp, pl, cl) are used to make the query shorter and easier to read instead of writing the full table names.
- The query also uses COUNT(sg.gamePlayerID) to calculate how many goals each player has scored, and GROUP BY ensures each player only appears once with their total.
- The results are sorted by the total number of goals in descending order, so the players with the most goals are shown at the top of the list.

4.2d. Provide a list of the goals your highest goal scorer scored. What game the goal(s) took place in, against which club and on which ground. List the goals in chronological order.

Syntax for query 4.2d

```
SELECT

pl.firstName || ' ' || pl.lastName AS "Player Name",

gm.gameID AS "Game Number",

gm.description AS "Fixture",

opp.clubName AS "Scored Against",

grn.name AS "Ground Name",

sg.gameMinute AS "Minute Scored",

gm.gamedate AS "Game Date"

FROM ScoredGoal sg

JOIN GamePlayer gp ON sg.gamePlayerID = gp.gamePlayerID

JOIN Player pl ON gp.playerID = pl.playerID

JOIN Game gm ON gp.gameID = gm.gameID

JOIN ClubGame cg ON gp.gameID = cg.gameID AND cg.clubID = pl.clubID
```

```
JOIN ClubGame oppCG ON oppCG.gameID = gm.gameID AND oppCG.clubID <> cg.clubID

JOIN Club opp ON opp.clubID = oppCG.clubID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE gp.playerID IN (

SELECT gp2.playerID

FROM ScoredGoal sg2

JOIN GamePlayer gp2 ON sg2.gamePlayerID = gp2.gamePlayerID

GROUP BY gp2.playerID

HAVING COUNT(sg2.goalID) = (

SELECT MAX(totalGoals)

FROM (

SELECT gp3.playerID, COUNT(sg3.goalID) AS totalGoals

FROM ScoredGoal sg3

JOIN GamePlayer gp3 ON sg3.gamePlayerID = gp3.gamePlayerID

GROUP BY gp3.playerID

    )

  )

)

ORDER BY gm.gamedate, gm.gameID, sg.gameMinute;
```

Screen-grab/Result of query 4.2d

Player Name	Game Number	Fixture	Scored Against	Ground Name	Minute Scored	Game Date
Glen Evans	GM01	CL01 v CL02	Dunsborough Dingos	Eagle Bay Centre	44	05/12/2025
Glen Evans	GM01	CL01 v CL02	Dunsborough Dingos	Eagle Bay Centre	66	05/12/2025
Glen Evans	GM01	CL01 v CL02	Dunsborough Dingos	Eagle Bay Centre	78	05/12/2025
Glen Evans	GM13	CL01 v CL05	Reinscourt Rabbits	Eagle Bay Centre	69	06/02/2025
Glen Evans	GM21	CL01 v CL07	New Caves Rd Numbats	Eagle Bay Centre	81	06/16/2025
Glen Evans	GM25	CL01 v CL08	Hanes Hawkes	Eagle Bay Centre	81	06/23/2025
Glen Evans	GM29	CL02 v CL01	Dunsborough Dingos	Quedjinup Reserve	85	06/30/2025
Glen Evans	GM37	CL04 v CL01	Mary-Brook Magpies	Kealy Reserve	36	07/15/2025
Glen Evans	GM37	CL04 v CL01	Mary-Brook Magpies	Kealy Reserve	52	07/15/2025

9 rows returned in 0.00 seconds Download

Explanation of query 4.2d

- The SELECT query has one main table which is ScoredGoal (sg).

- The query shows details such as the player name, game number, fixture description, opponent club, ground name, minute scored, and game date from the related tables.
- Other tables are joined to add more information:

  GamePlayer (gp) is joined to connect each scored goal to the player who scored it.

  Player (pl) is used to display the player's first name and last name together as the full name.

  Game (gm) is used to show details of the match, such as the game number, description, and date.

  ClubGame (cg) is joined with Player (pl) to identify the club that the scoring player belongs to.

  ClubGame (oppCG) and Club (opp) are used again to find the opponent club in that game.

  Ground (grn) is used to show the name of the ground where the game was played.

- This query first counts how many goals each player scored during the season (using the logic of the PlayerGoals table in the original version).
- Then, it finds the player or players who scored the most goals (the TopScorer step), by comparing each player's total goals to the maximum number of goals scored.
- Finally, the main query lists all the goals scored by these top scorer(s), showing the player's name, which game the goal was in, the fixture, the club they scored against, the ground where it happened, the minute of the goal, and the date of the game.
- Aliases (sg, gp, pl, gm, cg, cl, grn) are used to make the query shorter and easier to read instead of repeating full table names.
- The results are ordered by game date, game number, and minute scored, so the goals appear in the exact order they happened.

4.3 MVP (most valuable player) of the Season

4.3a. List all players who were given BoG-points during the season. Order the list from most BoG-points to Least BoG-points scored, and include their total number of BoG-points scored. This should tell you who won the Most Valuable Player in the South West Soccer Competition. You will score more SQL marks if you can somehow improve the result to show a summary of information such as how many times each player scored 3-points, or 2-points or 1-point as part of your whole presentation

Syntax for query 4.3a

```
SELECT

pl.playerID,

pl.firstName || ' ' || pl.lastName AS "Player Name",


/* Count how many times player scored 3 points */

(SELECT COUNT(*)

FROM BogPlayer bp3

JOIN BestOnGround bog3 ON bp3.bogID = bog3.bogID

JOIN GamePlayer gp3 ON bp3.gamePlayerID = gp3.gamePlayerID

WHERE gp3.playerID = pl.playerID AND bog3.points = 3) AS "Rank 1 (3 Points)",


/* Count how many times player scored 2 points */

(SELECT COUNT(*)

FROM BogPlayer bp2
```

```sql
JOIN BestOnGround bog2 ON bp2.bogID = bog2.bogID

JOIN GamePlayer gp2 ON bp2.gamePlayerID = gp2.gamePlayerID

WHERE gp2.playerID = pl.playerID AND bog2.points = 2) AS "Rank 2 (2 Points)",

/* Count how many times player scored 1 point */

(SELECT COUNT(*)

FROM BogPlayer bp1

JOIN BestOnGround bog1 ON bp1.bogID = bog1.bogID

JOIN GamePlayer gp1 ON bp1.gamePlayerID = gp1.gamePlayerID

WHERE gp1.playerID = pl.playerID AND bog1.points = 1) AS "Rank 3 (1 Point)",

/* Total BoG points */

(SELECT SUM(bog.points)

FROM BogPlayer bp

JOIN BestOnGround bog ON bp.bogID = bog.bogID

JOIN GamePlayer gp ON bp.gamePlayerID = gp.gamePlayerID

WHERE gp.playerID = pl.playerID) AS "Total BoG Points"

FROM Player pl

WHERE pl.playerID IN (

SELECT DISTINCT gp.playerID

FROM BogPlayer bp

JOIN GamePlayer gp ON bp.gamePlayerID = gp.gamePlayerID

)

ORDER BY "Total BoG Points" DESC;
```

Screen-grab/Result of query 4.3a

Results | Explain | Describe | Saved SQL | History

PLAYERID	Player Name	Rank 1 (3 Points)	Rank 2 (2 Points)	Rank 3 (1 Point)	Total BoG Points
PL810	Chris Brooks	4	1	1	15
PL110	Glen Evans	2	3	2	14
PL711	Alex Martin	3	2	1	14
PL609	Frank Hughes	2	3	1	13

....

PL112	Robert Tan	0	0	1	1
PL606	Stuart Coleman	0	0	1	1
PL503	Robert Tan	0	0	1	1
PL107	Peter Murray	0	0	1	1

60 rows returned in 0.05 seconds Download

Explanation of query 4.3a

- The main table in this query is Player (pl). From here, we take each player's ID and combine their first name and last name into one column called Player Name.
- The query then uses small extra searches (subqueries) to calculate more details about each player:

  We add together (SUM) all the Best-on-Ground (BoG) points that a player received across all games. This gives the Total BoG Points. It also counts how many times a player got 3 points, shown as Rank 1 (3 Points).

  We count how many times a player got 2 points, shown as Rank 2 (2 Points).

  We count how many times a player got 1 point, shown as Rank 3 (1 Point).

- Short names called aliases (pl, bp, bog, gp) are used so the query is easier to read instead of typing the full table names every time.
- The WHERE condition makes sure only players who actually received BoG points are included, so players with zero points don't show up.
- Finally, the results are sorted by Total BoG Points from highest to lowest. This means the player at the top of the list is the one with the most BoG points, which is the Most Valuable Player (MVP) of the season.

4.3b. Write a query that summarises your MVP's season. Include a way to display when 3-points, 2-points and/or 1-point were scored and in which games. List this in chronological/game order.

Syntax for query 4.3b

```
SELECT gm.gameID, gm.gameDate, pl.playerID, pl.firstName || ' ' || pl.lastName AS "Player Name", bog.points
AS "BoG Points"

FROM Player pl

JOIN GamePlayer gp ON pl.playerID = gp.playerID

JOIN BogPlayer bp ON gp.gamePlayerID = bp.gamePlayerID

JOIN BestOnGround bog ON bp.bogID = bog.bogID

JOIN Game gm ON gp.gameID = gm.gameID

WHERE pl.playerID = 'PL810' /* PL810: Chris Brooks, MVP from 4.3a */

ORDER BY gm.gameDate;
```

Screen-grab/Result of query 4.3b

Results | Explain | Describe | Saved SQL | History

GAMEID	GAMEDATE	PLAYERID	Player Name	BoG Points
GM08	05/25/2025	PL810	Chris Brooks	3
GM19	06/13/2025	PL810	Chris Brooks	1
GM22	06/18/2025	PL810	Chris Brooks	3
GM25	06/23/2025	PL810	Chris Brooks	2
GM50	08/06/2025	PL810	Chris Brooks	3
GM53	08/11/2025	PL810	Chris Brooks	3

6 rows returned in 0.00 seconds Download

Explanation of query 4.3b

- The SELECT query had one main table which is Player (pl).

- The SQL query retrieves the gameID, gameDate, playerID, firstName and lastName and Bog points from the related tables.

- JOIN between GamePlayer (gp) and BogPlayer (bp) links the player's game record with BoG entry.

- JOIN between BogPlayer (bp) and BestOnGround(bog) retrives BoG points.

-JOIN between Game(gm) and GamePlayer (gp) links each BoG entry with its corresponding game information(gameID, gameDate).

-WHERE condition specifies a particular playerID (PL810) which corresponds to MVP identified in 4.3a. This ensures that only the season data of MVP is shown.

-ORDER BY condition sorts the results by gameDate in ascending order. Thus, Bog points are shown across the season chronologically.

4.4 Miscellaneous fun stuff

4.4a Which club had the best "away" record (i.e., scored the most points, won and/or drew the most often when playing away from their home ground) and what was the record.

Syntax for query 4.4a

```
SELECT cl.clubID, cl.clubName,

/* SUM of Away point */

SUM(rs.points) AS "Total Away Points",


/* The count of Away Win */

(SELECT COUNT(*)

FROM ClubGame cg2

JOIN Result rs2 ON cg2.resultID = rs2.resultID

JOIN Game gm2 ON cg2.gameID = gm2.gameID

JOIN Ground grn2 ON gm2.groundID = grn2.groundID

WHERE cg2.clubID = cl.clubID

AND cg2.clubID != grn2.clubID

AND rs2.description = 'Away Win') AS "Away Wins",
```

```sql
/* The count of Away Draw */
(SELECT COUNT(*)
FROM ClubGame cg3
JOIN Result rs3 ON cg3.resultID = rs3.resultID
JOIN Game gm3 ON cg3.gameID = gm3.gameID
JOIN Ground grn3 ON gm3.groundID = grn3.groundID
WHERE cg3.clubID = cl.clubID
AND cg3.clubID != grn3.clubID
AND rs3.description = 'Away Draw') AS "Away Draws",

/* The count of Away Loss */
(SELECT COUNT(*)
FROM ClubGame cg4
JOIN Result rs4 ON cg4.resultID = rs4.resultID
JOIN Game gm4 ON cg4.gameID = gm4.gameID
JOIN Ground grn4 ON gm4.groundID = grn4.groundID
WHERE cg4.clubID = cl.clubID
AND cg4.clubID != grn4.clubID
AND rs4.description = 'Away Loss') AS "Away Losses"

FROM Club cl

JOIN ClubGame cg ON cl.clubID = cg.clubID
JOIN Result rs ON cg.resultID = rs.resultID
JOIN Game gm ON cg.gameID = gm.gameID
JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID != grn.clubID
GROUP BY cl.clubID, cl.clubName

/* Finding the highest away team score */
HAVING SUM(rs.points) = (
```

```
SELECT MAX(highest_pts)

FROM (SELECT cg.clubID, SUM(rs.points) AS highest_pts

FROM ClubGame cg

JOIN Result rs ON cg.resultID = rs.resultID

JOIN Game gm ON cg.gameID = gm.gameID

JOIN Ground grn ON gm.groundID = grn.groundID

WHERE cg.clubID != grn.clubID

GROUP BY cg.clubID)

);
```

Screen-grab/Result of query 4.4a

CLUBID	CLUBNAME	Total Away Points	Away Wins	Away Draws	Away Losses
CL06	Wonnerup Wombats	31	4	2	1

1 rows returned in 0.02 seconds Download

Explanation of query 4.4a

The SELECT query had one main table which is Club (cl).

The SQL query retrieves the clubID and clubName from the Club (cl) table.

Subqueries were used to calculate additional information for each club:

From ClubGame(cg) and Result(rs) to SUM the total away points as Total Away Points.

From ClubGame(cg2), Result(rs2), Game(gm2), Ground(grn2) to COUNT the number of Away Wins as Away Wins.

From ClubGame(cg3), Result(rs3), Game(gm3), Ground(grn3) to COUNT the number of Away Draws as Away Draws.

From ClubGame(cg4), Result(rs4), Game(gm4), Ground(grn4) to COUNT the number of Away Losses as Away Losses.

Using aliases cl, cg, rs, gm, grn for the main query and cg2,cg3,cg4, rs2,rs3,rs4, gm2,gm3,gm4, and grn2,grn3,grn4 for the subqueries makes the query shorter and easier to read instead of full table names.

WHERE condition inside each subquery makes sure that only away games are counted by checking cg.clubID != grn.clubID.

The GROUP BY clause groups the result by clubID and clubName to calculate total away performance per club.

HAVING clause compares each club's SUM of away points with the maximum away points, calculated from a subquery (alias highest_pts). This ensures that only the club with the best away record are shown.

4.4b Which home-and-away game had the largest crowd watching the game.

Syntax for query 4.4b

```
SELECT

gm.description AS "Game Fixture",

gm.numOfSpectators AS "Number of Spectators",

cl.clubName AS "Home Club",

gr.name AS "Ground Name",

gr.address AS "Ground Address",

gm.gameDate AS "Game Date",

gm.startTime AS "Start Time",

gm.endTime AS "End Time"

FROM Game gm

JOIN Ground gr ON gm.groundID = gr.groundID

JOIN CompetitionWeek cw ON gm.weekID = cw.weekID

JOIN Club cl ON gr.clubID = cl.clubID

ORDER BY gm.numOfSpectators DESC

FETCH FIRST 1 ROWS ONLY;
```

Screen-grab/Result of query 4.4b

Bottom Splitter in Describe Saved SQL History

Game Fixture	Number of Spectators	Home Club	Ground Name	Ground Address	Game Date	Start Time	End Time
CL03 v CL08	1980	Broadwater Bandicoots	Broadwater Bay Oval	Broadwater WA	06/13/2025	09/01/2025	09/01/2025

1 rows returned in 0.01 seconds Download

Explanation of query 4.4b

Using aliases **gm**, **gr**, **cw**, and **cl** for Game, Ground, CompetitionWeek, and Club respectively, the query joins Ground on **gm.groundID = gr.groundID**. This connects each game to its specific ground and establishes the home ground relationship. Then CompetitionWeek is joined on **gm.weekID = cw.weekID** to link games to their respective competition weeks, ensuring we're only looking at regular season home-and-away games. Finally, Club is joined on **gr.clubID = cl.clubID** to connect the ground to its home club, providing the home team information.

The query selects game details including the fixture description, spectator count, home club name, ground information, and game timing. The **ORDER BY gm.numOfSpectators DESC** command sorts all games in descending order based on the number of spectators, with the highest attendance first.

Finally, the **FETCH FIRST 1 ROWS ONLY** command retrieves only the top result, which represents the home-and-away game with the largest crowd watching the game.

4.4c List all the referees who awarded cards during the season and order them from being strictest to least.

Syntax for query 4.4c

```
SELECT re.firstName || ' ' || re.lastName AS "Referee Name",

COUNT(ac.awardedCardID) AS "Total Cards Awarded" FROM AwardedCard ac

JOIN GameReferee gr ON ac.gameRefereeID = gr.gameRefereeID

JOIN Referee re ON gr.refereeID = re.refereeID

GROUP BY re.firstName, re.lastName

ORDER BY COUNT(ac.awardedCardID) DESC;
```

Screen-grab/Result of query 4.4c

Bottom Splitter in Describe Saved SQL History

Referee Name	Total Cards Awarded
Michael Jones	5
Sarah Wilson	3
James Patel	3
William Taylor	2

4 rows returned in 0.01 seconds Download

Explanation of query 4.4c

Using aliases **ref**, **gr** and **ac** for **Referee**, **GameReferee** and **AwardedCard** respectively, joined **GameReferee** on

ac.gameRefereeID = **gr.gameRefereeID**. It joins AwardedCards with specific game and referee combination. Then Referee is joined on **ref.refereeID = gr.refereeID** to join game referee relationship to actual referee details.

The **GROUP BY** command creates separates groups for each referee and **COUNT** function counts the number of cards each referee awarded.

Finally, the result is ordered in descending order of count of awarded cards by each referee using **ORDER BY** command.

~ End of Assignment ~