**Проект 2.1 DDL конструкции для создания и модернизации таблиц. Правила РСУБД для определения консистентности данных**

**Упражнение 1 - Первые таблицы**

Давайте разработаем модель данных для хранения информации по баскетбольным матчам Евролиги!   
Модель данных будет включать в себя несколько таблиц, для хранения данных по стадионам, командам, участникам команд и проведенным матчам.

Создайте реляционную таблицу, которая будет описывать сущность “стадион” (арену) команды.   
Необходимые атрибуты для создания таблицы перечислены ниже. Пожалуйста укажите имя создаваемой таблицы как **arena**.

| **атрибут** | **тип данных** | **Описание** |
| --- | --- | --- |
| ID | INTEGER | первичный ключ (Primary Key) |
| NAME | VARCHAR | наименование арены |
| SIZE | INTEGER | максимальное количество зрителей |

Создайте реляционную таблицу, которая будет описывать сущность “команда”.   
Необходимые атрибуты для создания таблицы перечислены ниже. Пожалуйста укажите имя создаваемой таблицы как **team**.

| **атрибут** | **тип данных** | **Описание** |
| --- | --- | --- |
| ID | INTEGER | первичный ключ (Primary Key) |
| CITY | VARCHAR | домашний город команды |
| NAME | VARCHAR | наименование команды |
| COACH\_NAME | VARCHAR | имя тренера команды |
| ARENA\_ID | INTEGER | стадион команды |

Создайте реляционную таблицу, которая будет описывать сущность “игрок”.   
Необходимые атрибуты для создания таблицы перечислены ниже. Пожалуйста укажите имя создаваемой таблицы как **player**.

| **атрибут** | **тип данных** | **Описание** |
| --- | --- | --- |
| ID | INTEGER | первичный ключ (Primary Key) |
| NAME | VARCHAR | имя игрока |
| POSITION | VARCHAR | позиция игрока на поле |
| HEIGHT | NUMERIC | рост игрока |
| WEIGHT | NUMERIC | вес игрока |
| SALARY | NUMERIC | зарплата игрока |
| TEAM\_ID | INTEGER | команда игрока |

**Глава II**

**Упражнение 2 - Правила консистентности**

Для таблицы **arena** (созданную в задании №1) укажите следующие правила консистентности данных (CHECK CONSTRAINTs)

* Поле NAME должно принимать NOT NULL значение
* Поле SIZE должно принимать NOT NULL значение
* Значение по умолчанию для поля SIZE - это 100 зрителей

Для таблицы **team** (созданную в задании №1) укажите следующие правила консистентности данных (CHECK CONSTRAINTs)

* Поле NAME должно принимать NOT NULL значение
* Поле CITY должно принимать NOT NULL значение
* Поле COACH\_NAME должно принимать NOT NULL значение
* Поле ARENA\_ID должно принимать NOT NULL значение

Для таблицы **player** (созданную в задании №1) укажите следующие правила консистентности данных (CHECK CONSTRAINTs)

* Поле NAME должно принимать NOT NULL значение
* Поле POSITION должно принимать NOT NULL значение
* Поле HEIGHT должно принимать NOT NULL значение
* Поле WEIGHT должно принимать NOT NULL значение
* Поле SALARY должно принимать NOT NULL значение
* Поле TEAM\_ID должно принимать NOT NULL значение
* Рост игрока (HEIGHT) должен быть строго больше 0
* Вес игрока (WEIGHT) должен быть строго больше 0
* Зарплата игрока (SALARY) должна быть строго больше 0

**Глава III**

**Упражнение 3 - Определение ключей**

Для таблицы **arena** (созданную в задании №1) укажите следующие правила консистентности данных по уникальности значений

* Имя стадиона (NAME) должно быть уникальным (UNIQUE KEY)

Для таблицы **team** (созданную в задании №1) укажите следующие правила консистентности данных по уникальности значений

* Имя команды (NAME) должно быть уникальным (UNIQUE KEY)
* Имя тренера команды (COACH\_NAME) должно быть уникальным (UNIQUE KEY)

Для таблицы **player** (созданную в задании №1) укажите следующие правила консистентности данных по уникальности значений (UNIQUE KEY)

* Имя игрока (NAME) должно быть уникальным

Для таблицы **team** (созданную в задании №1) укажите следующие правила консистентности данных по определения внешнего ключа (FOREIGN KEY)

* Поле ARENA\_ID должно быть внешним ключом на поле ID таблицы arena с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом ограничения по каскадному удалению данных (ON DELETE RESTRICT)

Для таблицы **player** (созданную в задании №1) укажите следующие правила консистентности данных по определения внешнего ключа (FOREIGN KEY)

* Поле TEAM\_ID должно быть внешним ключом на поле ID таблицы team с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом ограничения по каскадному удалению данных (ON DELETE CASCADE)

**Глава IV**

**Упражнение 4 - Дополнительная таблица**

Создайте реляционную таблицу, которая будет описывать сущность “игра” между командами.   
Необходимые атрибуты для создания таблицы перечислены ниже. Пожалуйста укажите имя создаваемой таблицы как **game**.

| **атрибут** | **тип данных** | **Описание** |
| --- | --- | --- |
| ID | INTEGER | первичный ключ (Primary Key) |
| OWNER\_TEAM\_ID | INTEGER | команда хозяйка матча |
| GUEST\_TEAM\_ID | INTEGER | команда гостья матча |
| GAME\_DATE | DATE | дата игры |
| WINNER\_TEAM\_ID | INTEGER | команда победитель |
| OWNER\_SCORE | INTEGER | количество очков команды хозяйки |
| GUEST\_SCORE | INTEGER | количество очков команды гостьи |
| ARENA\_ID | INTEGER | стадион проведения матча |

**Глава V**

**Упражнение 5 - Дополнительные правила консистентности**

Для таблицы **game** (созданную в задании №4) укажите следующие правила консистентности данных (CHECK CONSTRAINTs)

* Поле OWNER\_TEAM\_ID должно принимать NOT NULL значение
* Поле GUEST\_TEAM\_ID должно принимать NOT NULL значение
* Поле GAME\_DATE должно принимать NOT NULL значение
* Поле WINNER\_TEAM\_ID должно принимать NOT NULL значение
* Поле OWNER\_SCORE должно принимать NOT NULL значение
* Поле GUEST\_SCORE должно принимать NOT NULL значение
* Поле ARENA\_ID должно принимать NOT NULL значение
* Количество очков по умолчанию для OWNER\_SCORE должно быть равным 0
* Количество очков по умолчанию для GUEST\_SCORE должно быть равным 0
* Значение поля OWNER\_SCORE должно быть не меньше 0
* Значение поля GUEST\_SCORE должно быть не меньше 0

Также для таблицы **game** укажите следующие правила консистентности данных по уникальности значений (UNIQUE KEY)

* Пара команд в матче (OWNER\_TEAM\_ID, GUEST\_TEAM\_ID) должна быть уникальным

Дополнительно для таблицы **game** укажите следующие правила консистентности данных по определения внешнего ключа (FOREIGN KEY)

* Поле OWNER\_TEAM\_ID должно быть внешним ключом на поле ID таблицы team с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом каскадного удаления данных (ON DELETE CASCADE)
* Поле GUEST\_TEAM\_ID должно быть внешним ключом на поле ID таблицы team с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом каскадного удаления данных (ON DELETE CASCADE)
* Поле WINNER\_TEAM\_ID должно быть внешним ключом на поле ID таблицы team с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом каскадного удаления данных (ON DELETE CASCADE)
* Поле ARENA\_ID должно быть внешним ключом на поле ID таблицы arena с правилом каскадного обновления данных (ON UPDATE CASCADE) и правилом каскадного удаления данных (ON DELETE CASCADE)

**Проект 2.2 Уникальные / первичные ключи. Ключ поиска данных. SQL: конструкции SELECT, FROM, WHERE**

## Глава I

## Упражнение 1 - Первые данные

Давайте зарегистрируем (INSERT) следующие стадионы в таблице **arena** со следующими значениями

| **ID** | **NAME** | **SIZE** |
| --- | --- | --- |
| 10 | Палау Блауграна | 8250 |
| 20 | Визинк-Центр - Паласио де Депортес | 15500 |
| 30 | УСК ЦСКА им. А.Я. Гомельского | 5000 |
| 40 | Пис энд Френдшип Стадиум | 12000 |
| 50 | СИБУР Арена | 7140 |

Давайте зарегистрируем (INSERT) следующие команды в таблице **team** со следующими значениями.   
Обратите внимание что необходимо сопоставить имя стадиона в таблице ниже с идентификатором стадиона из таблицы **arena**

| **ID** | **CITY** | **NAME** | **COACH\_NAME** | **ARENA** |
| --- | --- | --- | --- | --- |
| 10 | Барселона | Барселона | Шарунас Ясикявичюс | Палау Блауграна |
| 20 | Мадрид | Реал Мадрид | Пабло Ласо | Визинк-Центр - Паласио де Депортес |
| 30 | Москва | ЦСКА | Димитрис Итудис | УСК ЦСКА им. А.Я. Гомельского |
| 40 | Пирей | Олимпиакос | Георгиос Барцокас | Пис энд Френдшип Стадиум |
| 50 | Санкт-Петербург | Зенит | Хавьер Паскуаль | СИБУР Арена |

Давайте зарегистрируем (INSERT) следующих игроков в таблице **player** со следующими значениями.   
Обратите внимание что необходимо сопоставить имя команды в таблице ниже с идентификатором команды из таблицы **team**

| **ID** | **NAME** | **POSITION** | **HEIGHT** | **WEIGHT** | **SALARY** | **TEAM** |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | Рафа Вильяр | защитник | 188 | 85 | 100000 | Барселона |
| 20 | Кайл Курич | защитник | 193 | 85 | 100000 | Барселона |
| 30 | Ибу Дьянко Баджи | центровой | 211 | 103 | 200000 | Барселона |
| 40 | Ник Калатес | разыгрывающий | 198 | 97 | 150000 | Барселона |
| 50 | Никола Миротич | форвард | 208 | 107 | 175000 | Барселона |
| 60 | Джейси Кэрролл | защитник | 188 | 82 | 175000 | Реал Мадрид |
| 70 | Эли Джон Ндиайе | центровой | 203 | 110 | 275000 | Реал Мадрид |
| 80 | Уолтер Тавареш | центровой | 220 | 120 | 273000 | Реал Мадрид |
| 90 | Томас Давид Эртель | разыгрывающий | 189 | 88 | 173000 | Реал Мадрид |
| 100 | Гершон Ябуселе | форвард | 203 | 118 | 99000 | Реал Мадрид |
| 110 | Габриэль Иффе Лундберг | защитник | 193 | 96 | 101000 | ЦСКА |
| 120 | Юрий Умрихин | защитник | 190 | 75 | 251000 | ЦСКА |
| 130 | Иван Анатольевич Ухов | разыгрывающий | 193 | 77 | 175000 | ЦСКА |
| 140 | Александр Хоменко | разыгрывающий | 192 | 85 | 375000 | ЦСКА |
| 150 | Андрей Лопатин | лёгкий форвард | 208 | 92 | 205000 | ЦСКА |
| 160 | Тайлер Дорси | защитник | 193 | 83 | 205000 | Олимпиакос |
| 170 | Яннулис Ларенцакис | защитник | 196 | 87 | 75000 | Олимпиакос |
| 180 | Хассан Мартин | центровой | 201 | 107 | 375000 | Олимпиакос |
| 190 | Михалис Лунцис | разыгрывающий | 195 | 90 | 475000 | Олимпиакос |
| 200 | Георгиос Принтезис | форвард | 205 | 104 | 105000 | Олимпиакос |
| 210 | Билли Джеймс Бэрон | защитник | 188 | 88 | 75000 | Зенит |
| 220 | Артурас Гудайтис | центровой | 208 | 99 | 165000 | Зенит |
| 230 | Денис Захаров | разыгрывающий | 192 | 88 | 163000 | Зенит |
| 240 | Миндаугас Кузминскас | лёгкий форвард | 204 | 93 | 295000 | Зенит |
| 250 | Алекс Пойтресс | форвард | 201 | 108 | 247000 | Зенит |

## Глава II

## Упражнение 2 - Оперативные данные

Давайте зарегистрируем (INSERT) следующие проведенные матчи в таблице **game** со следующими значениями.   
Обратите внимание что необходимо сопоставить имя команды хозяйки, гостьи, победителя в таблице ниже с идентификатором (id) команды из таблицы **team**, более того необходимо сопоставить название стадиона из таблицы ниже с идентификатором стадиона из таблицы **arena**

| **ID** | **OWNER\_TEAM** | **GUEST\_TEAM** | **GAME\_DATE** | **WINNER\_TEAM** | **OWNER\_SCORE** | **GUEST\_SCORE** | **ARENA** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | Барселона | Зенит | 2021-10-22 | Барселона | 84 | 58 | Палау Блауграна |
| 20 | Барселона | ЦСКА | 2021-11-17 | Барселона | 81 | 73 | Палау Блауграна |
| 30 | Барселона | Реал Мадрид | 2021-10-12 | Барселона | 93 | 80 | Палау Блауграна |
| 40 | Барселона | Олимпиакос | 2021-10-15 | Барселона | 83 | 68 | Палау Блауграна |
| 50 | Зенит | Реал Мадрид | 2022-12-15 | Реал Мадрид | 68 | 75 | СИБУР Арена |
| 60 | Зенит | ЦСКА | 2022-01-15 | ЦСКА | 67 | 77 | СИБУР Арена |
| 70 | Зенит | Олимпиакос | 2022-10-20 | Зенит | 84 | 78 | СИБУР Арена |
| 80 | Реал Мадрид | ЦСКА | 2021-10-28 | Реал Мадрид | 71 | 65 | Визинк-Центр - Паласио де Депортес |
| 90 | Реал Мадрид | Олимпиакос | 2022-02-01 | Реал Мадрид | 75 | 67 | Визинк-Центр - Паласио де Депортес |
| 100 | ЦСКА | Олимпиакос | 2022-02-02 | ЦСКА | 79 | 78 | УСК ЦСКА им. А.Я. Гомельского |

## Глава III

## Упражнение 3 - Первый запрос

Напишите SQL запрос который возвращает имена стадионов (поле NAME) , вместимость которых строго больше 9000.   
Результат отсортируйте по имени стадиона в прямом порядке

## Глава IV

## Упражнение 4 - Второй запрос

Напишите SQL запрос который возвращает имена (поле NAME) всех “защитников” и “форвардов”.   
Результат отсортируйте по имени игрока в обратном порядке

## Глава V

## Упражнение 5 - Третий запрос

Напишите SQL запрос который возвращает имена игроков (поле NAME) ростом больше 215 см или весом больше 120 кг.   
Результат отсортируйте по имени игрока в прямом порядке.

**Проект 2.3 Конструкция ORDER BY. Использование формул и атрибутов. Вложенные запросы в SELECT, FROM, WHERE конструкциях**

## Глава I

## Упражнение 1 - Поиск игроков с условием

Напишите SQL запрос который возвращает имена игроков и их позицию на площадке , рост которых лежит в диапазоне от 188 до 200 см и зарплата которых лежит в диапазоне от 100000 до 150000. Пожалуйста используйте конструкцию BETWEEN.   
Результат отсортируйте по имени в убывающем порядке

## Глава II

## Упражнение 2 - Конкатенация данных

Напишите SQL запрос который возвращает данные по игровым командам в следующем виде (один столбец со всеми данными внутри) с сортировкой по столбцу “полная информация”.   
Обратите внимание на пример данных ниже, укажите такие же имена столбцов в результате вашего запроса.

| **полная информация** |
| --- |
| город: Барселона; команда: Барселона; тренер: Шарунас Ясикявичюс |
| город: Мадрид; команда: Реал Мадрид; тренер: Пабло Ласо |
| город: Москва; команда: ЦСКА; тренер: Димитрис Итудис |
| город: Пирей; команда: Олимпиакос; тренер: Георгиос Барцокас |
| город: Санкт-Петербург; команда: Зенит; тренер: Хавьер Паскуаль |

## Глава III

## Упражнение 3 - Поиск по списку

Напишите SQL запрос который возвращает список стадионов и их вместимости если идентификатор стадиона должен быть выбран из списка 10, 30, 50.   
Пожалуйста примените конструкцию **IN**.   
Результат запроса отсортируйте по вместимости стадиона И затем по его наименованию.

## Глава IV

## Упражнение 4 - Отрицание поиска по списку

Напишите SQL запрос который возвращает список стадионов и их вместимости если идентификатор стадиона НЕ должен соответствовать значениям из списка 10, 30, 50.   
Результат запроса отсортируйте по вместимости стадиона И затем по его наименованию.

## Глава V

## Упражнение 5 - Поиск по диапазону

Напишите SQL запрос который возвращает имена игроков и их позицию на площадке , рост которых лежит в диапазоне от 188 до 220 см. и позиция игроков должны быть “центровой” или “защитник”.   
Пожалуйста используйте конструкцию **BETWEEN** и **IN**.   
Результат отсортируйте по позиции и по имени в убывающем порядке для двух полей.   
Обратите внимание на пример данных ниже, укажите такие же имена столбцов в результате вашего запроса.

| **имя игрока** | **позиция на площадке** |
| --- | --- |
| Эли Джон Ндиайе | центровой |
| ... | ... |

**Проект 2.4 Конструкции INSERT / UPDATE / DELETE. Работа с множествами: UNION, INTERSECT, EXCEPT**

## Глава I

## Упражнение 1 - Создание общего списка данных

Напишите SQL запрос, который возвращает имена стадионов и имена команд в одном списке.   
Результат отсортируйте в убывающем порядке по алфавиту. Пример части результата представлен ниже

| **name** |
| --- |
| ЦСКА |
| УСК ЦСКА им. А.Я. Гомельского |
| СИБУР Арена |
| Реал Мадрид |
| Пис энд Френдшип Стадиум |
| ... |

## Глава II

## Упражнение 2 - Создание общего списка данных с указанием категории

Напишите SQL запрос, который возвращает имена стадионов и имена команд в одном списке, но с типом что запись является или “стадион” или “команда”.   
Результат отсортируйте в убывающем порядке по типу записи (object\_type) и потом по имени в возрастающем порядке.   
Пример части результата представлен ниже

| **name** | **object\_type** |
| --- | --- |
| Визинк-Центр - Паласио де Депортес | стадион |
| Палау Блауграна | стадион |
| ... | ... |
| Барселона | команда |
| ... | ... |

## Глава III

## Упражнение 3 - Персонализированная сортировка данных

Напишите SQL запрос, который возвращает имена и зарплаты игроков в отсортированном списке по зарплатам в возрастающем порядке, НО зарплата со значением 475 000 должна быть на первом месте.   
Ограничьте вывод данных 5 строками.   
Пример результата представлен ниже

| **name** | **salary** |
| --- | --- |
| Михалис Лунцис | 475000 |
| Яннулис Ларенцакис | 75000 |
| Билли Джеймс Бэрон | 75000 |
| Гершон Ябуселе | 99000 |
| Рафа Вильяр | 100000 |

**Подсказка**: используйте в **ORDER BY** синтаксический элемент CASE … END. Саму конструкцию можно посмотреть вот [здесь](https://postgrespro.ru/docs/postgrespro/14/functions-conditional)

## Глава IV

## Упражнение 4 - Разница между множествами

Напишите SQL запрос, который возвращает разницу по полю ID между таблицами игроков и команд.   
Результат отсортируйте по полю ID и выведите первые 10 строк.   
Пример части результата представлен ниже

| **id** |
| --- |
| 60 |
| 70 |
| 80 |
| ... |

## Глава V

## Упражнение 5 - Симметрическая разница между множествами

Напишите SQL запрос который подсчитывает симметрическую разницу множеств A и B.

(A \ B) ⋃ (B \ A)

где A - таблица **arena**, B - таблица **game**, “\” - это разница множеств , “⋃” - объединение множеств .   
Необходимо подсчитать на основании атрибута id из обеих таблиц.   
Результат отсортируйте по 1 столбцу. Пример вывода результата представлен ниже.

| **id** |
| --- |
| 60 |
| 70 |
| 80 |
| 90 |
| 100 |