

Министерство образования и науки Российской Федерации  
**ФГАОУ ВО «РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**  
**(RUDN University )**

Инженерная академия  
Институт космических технологий  
Департамент механики и мехатроники

## **КУРСОВАЯ РАБОТА**

**По дисциплине:** Системы управления базами данных

**По теме:** Информационная система военного округа

Выполнил: ст-т. гр. ИУСбд-01-17

Кормилицын Артем

Проверила: ст. Преподаватель

Баландина Г.И.

Дата:

Подпись:

Москва, 2020

## **Содержание**

Введение.....	3
Исследование предметной области.....	9
Создание базы данных.....	11
SQL запросы .....	15
Выходы.....	26
Литература .....	27
Приложение.....	28

## **Введение**

*The most important motivation for the research work  
that resulted in the relational model was the objective of providing a strict  
and clear boundary between the logical and physical aspects of database management.*

**Эдгар Франк Кодд**

В современный век информации невозможно представить сложные технологические механизмы без информационных систем. Человек сталкивается с информационными системами, когда использует социальные сети, делает телефонные звонки, резервирует номера в гостинице или авиабилеты, или когда оплачивает покупки в Интернет магазине, примеров огромное множество. Но что такое информационная система (ИС)?

Это сложный многофункциональный программно-аппаратный комплекс, который обеспечивает выполнение следующих функций:

- 1) Надежное хранение информации в памяти компьютера;
- 2) Выполнение преобразований информации и вычислений, реализующих функции бизнес-логики;
- 3) ИС представляет пользователю удобный и легко осваиваемый интерфейс, в котором можно вводить информацию и получать результат.

В отличии от фундаментальных наук, например, математика, физика или философия, истории которых исчисляются веками, история информационных систем включает в себя несколько десятилетий.

В 60-е годы информация, полученная от информационных систем, стала применяться для периодической отчетности уже по множеству параметров. В 70-е годы предприятия начинают использоваться информационные системы в качестве средства управления производством, которое могло поддержать и ускорить процесс подготовки и принятия решений. В большинстве своем информационные системы того времени создавались для решения определенного круга задач, которые четко определялись на начальном этапе и не изменялись в процессе жизни системы. В эти же годы возникают первые персональные компьютеры (ПК), которые становятся доступными не только крупным корпорациям и отдельным организациям, но и обычным людям.

Для ПК разрабатывается множество различных информационных систем, и базы данных (БД) создаются для решения разных взаимосвязанных задач, когда данные поставляются из различных источников.

Основные черты ИС того поколения:

- 1) В основе информационных систем лежат базы данных;
- 2) Все программное обеспечение ИС делится на составные части, или крупные модули;
- 3) Модули решают прикладные задачи (бизнес логику), специальная часть, которая обслуживает исключительно хранение и обработку данных.

Большие ЭВМ – это основные системы того времени. В во второй половине XX века развивалось множество языков программирования. Это были языки достаточного высокого уровня, также разрабатывались языки и для баз данных. Что касается архитектуры информационных систем, то в тот момент были популярны две модели:

- 1) Модель локальной информационной системы, которая размещена на ПК, и обычно ею пользовался один конкретный пользователь
- 2) Многопользовательская система, многопользовательские версии, расположенные на больших ЭВМ, пользователи в свою очередь подключаются в терминальном режиме.

Что привело к интенсивному развитию баз данных?

В первую очередь, это понимание, что данные важнее, чем программы, которые их обрабатывают. Ранее считалось, что компьютер в основном создан для выполнения числовых расчетов. Любая программа получала какие-то данные на входе, обрабатывала эти данные, выдавала результат и заканчивалась. Но жизнь заканчивалась вместе с окончанием работы программы. А потом появилось ощущение, что данные должны жить долго, должны быть надежно защищены, к ним нужно организовать доступ множества пользователей, а программы как раз могут меняться, могут исчезать, появляться новые, но на первый план выходит именно хранение и обработка данных. Следующая причина — это появление мощных вычислительных систем. потому что для обработки больших объемов данных требуются вычислительные мощности.

Вспомним эмпирический закон Гордона Мура, одного из основателей компании Intel, который сказал, что производительность любого компьютера удваивается каждые два года. Также причиной интенсивного развития баз данных был рост объемов носителей информации и изменение их типа. До 1960 года устройства для хранения данных были последовательными. Это были барабаны или ленты, требовалось перематывать эти ленты до определенного места, и после этого считывать данные. Позже появились устройства прямого доступа к данным, что позволило совершать выборочные обновления и организовывать хранение достаточно сложных структур данных. Рост объемов носителей

информации. Можно вспомнить фразу, неосторожно оброненную Биллом Гейтсом в 1981 году. Он сказал, что 640 килобайт должно хватить всякому. Но мы сейчас понимаем, что такого объема недостаточно даже для сохранения простой фотографии. И сейчас диск объемом в 500 гигабайт уже не кажется нам большим. Также причиной распространения баз данных стало развитие сети Интернет, которая появилась в 1969 году, называлась она тогда ARPANET. Но в поздний период это была уже огромная информационная сеть, в которой непрерывно циркулировали огромные объемы данных. Пользователи получили доступ к этой информации и могли ее использовать для решения различных задач. Развитие информационных систем и баз данных способствовало также развитию языков программирования высокого уровня, которые создавались в частности и для управления данными. И последнее, что стоит упомянуть, это методы искусственного интеллекта. Пионером в этой области можно считать Алана Тьюринга, который в 1950 году написал статью под названием «Может ли машина мыслить?».

Что же такое Базы данных? Базы данных появляются, когда нам нужно хранить и обрабатывать большие объемы информации. Информация хранится и обрабатывается в вычислительной системе. Кроме больших объемов информация должна быть хорошо структурирована. То есть в ней должны быть выделены составные части, и прослеживаться связи между этими данными. И структура базы данных должна быть такой, чтобы обеспечивать нам удобный поиск и обработку этих данных.

Этапы развития архитектуры баз данных:

- I) Развитие БД на больших ЭВМ – этап появления первых СУБД;
- II) СУБД, разработанные для персональных компьютеров
- III) Распределенные базы данных

БД на больших ЭВМ, в качестве основы для хранения данных использовали обычные файлы.

С точки зрения прикладной программы, файл – это именованная область во внешней памяти, в которой можно записывать информацию и из которой можно эту информацию считывать. Каждая программа, работающая с файлом, должна была внутри себя хранить структуру данных, которая соответствовала структуре этого файла. При попытке изменений структуры файла требовалось перезаписывать или пересоздавать заново работающие с этой информацией программы. В эпоху больших ЭВМ компанией IBM был инициирован исследовательский проект, по исследованию возможностей реляционной базы данных, и на основе этого исследования была создана первая реляционная СУБД, которая называлась System R. Для этой реляционной модели данных появляются языки программирования высокого уровня, направленные непосредственно на работу с

данными. В эпоху персональных СУБД появлялось множество программ, предназначенных для работы неподготовленных пользователей. Эти программы были очень просты и интуитивно понятны. Это были текстовые редакторы, электронные таблицы и другие файлы с понятной и ясной структурой, с удобным интерфейсом, который позволял пользователям вводить информацию и формировать какие-то отчеты.

Появление множества программ сказалось и на базах данных, которые разрабатывались для персональных компьютеров и позволяли хранить достаточно большие объемы информации. Доступность персональных компьютеров дала возможность пользователям из многих областей знаний, которые ранее не использовали компьютеры в своей деятельности, обратиться к ним.

И спрос на развитые и удобные программы по обработке данных заставил поставщиков ПО разрабатывать и создавать все новые и новые системы. Эти системы принято называть настольными СУБД, или десктопными моделями. История развивается по спирали, поэтому от процесса персонализации пошел обратный процесс — интеграции. Множится количество локальных сетей, развиваются глобальные сети, и все больше информации передается между компьютерами. Эта информация может быть доступна различным пользователям для решения различных прикладных задач. И сейчас остро встает вопрос согласованности данных, которые обрабатываются различными пользователями. И успешное решение этих задач привело в появлению распределенных СУБД, которые сохраняют все преимущества ранее существовавших настольных СУБД, и в то же время позволяют организовывать параллельную обработку информации и надежную поддержку целостности в базах данных.

## **Задание к курсовой работе**

### Информационная система военного округа

Военные части округа расквартированы по различным местам дислокации, причем в одном месте могут располагаться несколько частей. Каждая воинская часть состоит из рот, роты из взводов, взводы из отделений, в свою очередь воинские части объединяются в дивизии, корпуса или бригады, а те в армии. Военный округ представлен офицерским составом (генералы, полковники, подполковники, майоры, капитаны, лейтенанты) и рядовым и сержантским составом (старшины, сержанты, прапорщики, ефрейторы, рядовые). Каждая из перечисленных категорий военнослужащих может иметь характеристики, присущие только этой категории: для генералов это может быть дата окончания академии, дата присвоения генеральского звания и т.д. Каждое из подразделений имеет командира, причем военнослужащие офицерского состава могут командовать любым из вышеперечисленных подразделений, а военнослужащие рядового и сержантского состава только взводом и отделением. Все военнослужащие имеют одну или несколько воинских специальностей.

Каждой воинской части придана боевая и транспортная техника: БМП, тягачи, автотранспорт и пр. и вооружение: карабины, автоматическое оружие, артиллерия, ракетное вооружение и т.д. Каждая из перечисленных категорий боевой техники и вооружения также имеет специфические, присущие только ей атрибуты и по каждой категории может быть несколько видов техники и вооружения. Инфраструктура военной части представлена набором сооружений (сооружение №1, сооружение №2 ...), некоторые из которых предназначены для дислокации подразделений части.

#### **Виды запросов в информационной системе:**

1. Получить перечень всех частей военного округа, указанной армии, дивизии, корпуса и их командиров.
2. Получить данные по офицерскому составу в целом и по офицерскому составу указанного звания всех частей военного округа, отдельной армии, дивизии, корпуса, военной части.
3. Получить данные по рядовому и сержантскому составу в целом и с учетом указанного звания всех частей военного округа, отдельной армии, дивизии, корпуса, военной части.
4. Получить цепочку подчиненности снизу доверху для указанного военнослужащего.
5. Получить перечень мест дислокации всех частей военного округа, отдельной армии,

дивизии, корпуса, военной части.

6. Получить данные о наличии боевой технике в целом и с учетом указанной категории или вида во всех частях военного округа, в отдельной армии, дивизии, корпусе, военной части.

7. Получить перечень сооружений указанной военной части, перечень сооружений, где дислоцировано более одного подразделения, где не дислоцировано ни одного подразделения.

8. Получить перечень военных частей, в которых число единиц указанного вида боевой техники больше 5 (нет указанной боевой техники).

9. Получить данные о наличии вооружения в целом и с учетом указанной категории или вида во всех частях военного округа, в отдельной армии, дивизии, корпусе, военной части.

10. Получить перечень военных специальностей, по которым в округе, в отдельной армии, дивизии, корпусе, военной части более пяти специалистов (нет специалистов).

11. Получить перечень военнослужащих указанной специальности в округе, в отдельной армии, дивизии, корпусе, военной части, в указанном подразделении некоторой военной части.

12. Получить перечень военных частей, в которых число единиц указанного вида вооружения больше 10 (нет указанного вооружения).

13. Получить данные об армии, дивизии, корпусе, в которые входит больше всего (меньше всего) военных частей.

## **Исследование “mini – world”**

В задании к курсовой работе подробно описано как устроен mimi-world в предлагаемой базе данных.

Основная цель создания информационной системы военного округа – это оперативный поиск и редактирования информации о военных частях, оборудовании т.д. Воспользуемся концепцией моделирования модели сущности-отношения (ER), которая является популярной концептуальной моделью данных высокого уровня.

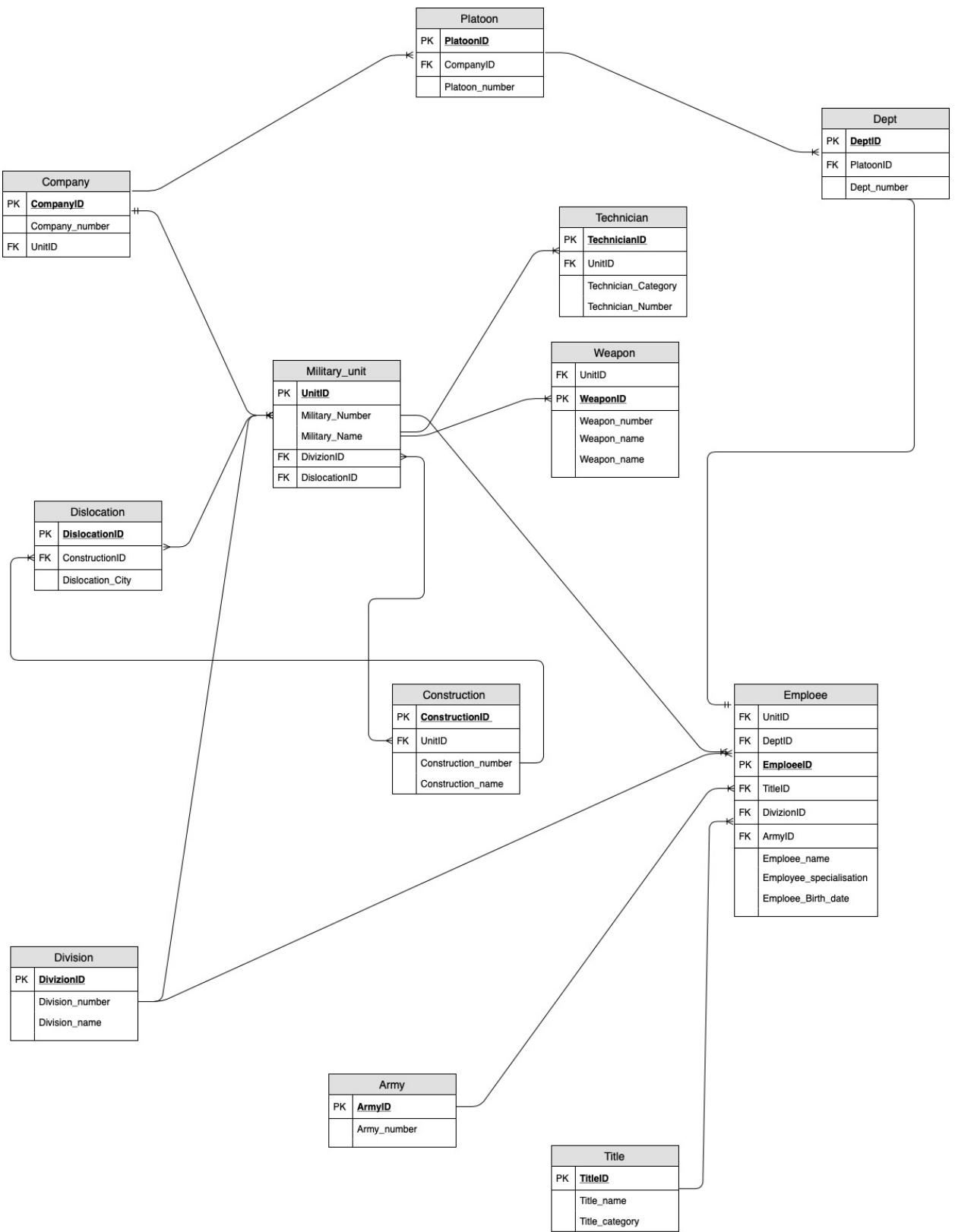
Сущностями нашей модели будут следующие объекты:

Военная часть	<b>Military_unit</b>	Дислокация	<b>Dislocation</b>
Взвод	<b>Platoon</b>	Вооружение	<b>Weapon</b>
Рота	<b>Company</b>	Построение	<b>Construction</b>
Дивизион	<b>Divizion</b>	Звание	<b>Title</b>
Армия	<b>Army</b>		
Отдел	<b>Dept</b>		
Сотрудник	<b>Employee</b>		
Техник	<b>Technician</b>		

Построим ER-диаграмму в сервисе draw.io. Данный сервис был выбран из некоторых объективных причин: доступность браузерной версии (Рис.3 – Приложение);

- 1) Удобный интуитивно понятный пользовательский интерфейс;
- 2) Возможность экспорта файла с различными расширениями;
- 3) Интеграция с такими сервисами разработки, как GitHub, GitLab, Trello.

Полученная диаграмма изображена на Рис. 1.



**Рис. 1** ER-диаграмма информационной системы военного округа

## Создание базы данных

При создании базы данных воспользуемся средой Oracle worksheet на платформе <https://livesql.oracle.com> (Рис. 4 – Приложение).

Листинг создания БД «Информационная система военного округа»:

```
CREATE TABLE Army
(ArmyID integer primary key not null,
Numb integer not null);
INSERT INTO Army(ArmyID,Numb) VALUES (1,1);
INSERT INTO Army(ArmyID,Numb) VALUES (2,2);
INSERT INTO Army(ArmyID,Numb) VALUES (3,3);

CREATE TABLE Division
(DivisionID integer primary key not null,
Numb integer not null,
Name char (50) not null);
INSERT INTO Division(DivisionID,Numb,Name) VALUES (1,1,'Пограничники');
INSERT INTO Division(DivisionID,Numb,Name) VALUES (2,2,'ВДВ');
INSERT INTO Division(DivisionID,Numb,Name) VALUES (3,3,'Военно-космические');

CREATE TABLE Title
(TitleID integer primary key not null,
Name char (50) not null,
Title_Category char (50) not null,
reit integer);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (1,'Рядовой','Рядовой состав',1);
INSERT INTO Title(TitleID,Name,Title_Category,reit) values (2,'Старший сержант','Сержантский
состав',4);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (3,'Ефрейтор','Сержантский состав',2);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (4,'Капитан','Офицер',8);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (5,'Лейтенант','Офицер',7);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (6,'Майор','Офицер',9);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (7,'Прaporщик','Сержантский состав',5);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (8,'Сержант','Сержантский состав',3);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (9,'Старший прaporщик','Сержантский
состав',6);
INSERT INTO Title(TitleID,Name,Title_Category,reit) VALUES (10,'Рядовой','Рядовой состав',1);

CREATE TABLE Military_unit
(UnitID integer primary key not null,
DivisionID integer references Division (DivisionID),
Numb integer not null,
Name char (50) not null);
alter table Military_unit
add (DislocationID integer references Dislocation (DislocationID));
INSERT INTO Military_unit(UnitID,DivisionID,Numb,DislocationID,Name) VALUES (1,1,1,1,'Моторизовання');
INSERT INTO Military_unit(UnitID,DivisionID,Numb,DislocationID,Name) VALUES (2,2,2,2,'Танковая');
INSERT INTO Military_unit(UnitID,DivisionID,Numb,DislocationID,Name) VALUES (3,3,3,1,'BBC');

CREATE TABLE Construction
```

```

(ConstructionID integer primary key not null,
UnitID integer references Military_unit (UnitID),
Numb integer not null,
Name char (50) not null);
INSERT INTO Construction(ConstructionID,UnitID,Numb,Name) VALUES (1,1,1,'Группа армии центр');
INSERT INTO Construction(ConstructionID,UnitID,Numb,Name) VALUES (2,2,2,'Группа армии север');
INSERT INTO Construction(ConstructionID,UnitID,Numb,Name) VALUES (3,3,3,'Группа армии юг');
INSERT INTO Construction(ConstructionID,UnitID,Numb,Name) VALUES (4,1,4,'Группа армии Кубань');
INSERT INTO Construction(ConstructionID,UnitID,Numb,Name) VALUES (5,2,1,'Группа армии Кавказ');

CREATE TABLE Dislocation
(DislocationID integer primary key not null,
ConstructionID integer references Construction (ConstructionID),
Dislocation_City char (50) not null);
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (1,1,'Краснодар');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (2,2,'Уфа');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (3,3,'Краснодон');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (4,1,'Севастополь');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (5,2,'Петергоф');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (6,3,'Санкт-Петербург');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (7,4,'Калининград');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (8,1,'Новоросийск');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (9,2,'Северск');
INSERT INTO Dislocation(DislocationID,ConstructionID,Dislocation_City) VALUES (10,4,'Улан-Удэ');

CREATE TABLE Tehnician
(TehnicianID integer primary key not null,
UnitID integer references Military_unit (UnitID),
Numb integer not null, Name char (50) not null);
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (1,1,1,'ДДТ');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (2,2,2,'Грузовик');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (3,3,3,'Танк Т-34');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (4,1,2,'Грузовик');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (5,1,3,'Танк Т-34');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (6,2,1,'Платон');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (7,2,3,'Танк Т-62');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (8,3,2,'Грузовик');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (9,3,1,'Козак');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (10,1,1,'Козак-М2');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (11,1,1,'Козак-М2');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (12,1,1,'Козак-М1');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (13,1,1,'Козак');
INSERT INTO Tehnician(TehnicianID,UnitID, Numb,Name) VALUES (14,1,1,'Оплот');

CREATE TABLE Weapon
(WeaponID integer primary key not null,
UnitID integer references Military_unit (UnitID),
Numb integer not null,
Name char (50) not null);
INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (1,1,1,'Ракетное вооружение');
INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (2,2,2,'Артилерия');

```

```

INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (3,3,3,'Автоматическое оружие');
INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (4,3,1,'Ракетное вооружение');
INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (5,1,2,'Артилерия');
INSERT INTO Weapon(WeaponID,UnitID,Numb,Name) VALUES (6,2,3,'Автоматическое оружие');

CREATE TABLE Company
(CompanyID integer primary key not null,
UnitID integer references Military_unit (UnitID),
Company_number integer not null);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (1,1,1);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (2,2,2);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (3,3,3);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (4,1,3);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (5,2,1);
INSERT INTO Company(CompanyID,UnitID,Company_number) VALUES (6,3,2);

CREATE TABLE Platoon
(PlatoonID integer primary key not null,
Numb integer not null,
CompanyID integer references Company (CompanyID));
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (1,1,1);
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (2,2,2);
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (3,3,3);
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (4,4,4);
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (5,5,1);
INSERT INTO Platoon(PlatoonID,Numb,CompanyID) VALUES (6,6,3);

CREATE TABLE Dept
(DeptID integer primary key not null,
PlatoonID integer references Platoon (PlatoonID),
Numb integer not null);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (1,1,1);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (2,1,2);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (3,3,3);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (4,4,4);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (5,5,2);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (6,6,3);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (7,1,1);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (8,2,2);
INSERT INTO Dept(DeptID,PlatoonID,Numb) VALUES (9,3,3);

CREATE TABLE Employee
(EmployeeID integer primary key not null,
Name char (150) not null,
Employee_Birth_date date,
TitleID integer references Title (TitleID),
UnitID integer references Military_unit (UnitID),
DivisionID integer references Division (DivisionID),
ArmyID integer references Army (ArmyID),
DeptID integer references Dept (DeptID),

```

```

Employee_specialisation char (150) not null;
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (1,'Кормилицын А. Ю',TO_DATE ('15-09-1999', 'DD-MM-YYYY'),1,1,1,1,1,'Директор
продовольственного цеха');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (2,'Узунова Е. Ф',TO_DATE ('17-08-1999', 'DD-MM-YYYY'),1,1,1,1,2,'Переводчик');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (3,'Вураки И. В',TO_DATE ('03-12-2000', 'DD-MM-YYYY'),2,2,2,2,3,'Инженер');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (4,'Гасанова З. А',TO_DATE ('21-05-1999', 'DD-MM-YYYY'),4,3,2,1,1,'Уборщик');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (5,'Иванова А. С',TO_DATE ('25-10-2001', 'DD-MM-YYYY'),6,1,1,1,6,'Преподаватель военного
дела');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (6,'Петров В. И',TO_DATE ('12-11-1993', 'DD-MM-YYYY'),7,3,3,3,4,'Инженер');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (7,'Волков Я. А',TO_DATE ('05-03-1998', 'DD-MM-YYYY'),5,2,1,2,5,'Садовник');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (8,'Зубков А. Д',TO_DATE ('09-01-1965', 'DD-MM-YYYY'),8,2,3,1,4,'Лаборант');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (9,'Манандин Г. И',TO_DATE ('16-07-1979', 'DD-MM-YYYY'),10,1,2,3,9,'Инженер');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (10,'Герман А. А',TO_DATE ('22-04-1987', 'DD-MM-YYYY'),8,2,3,1,4,'Инструктор');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (11,'Максаков И. П',TO_DATE ('28-10-1973', 'DD-MM-YYYY'),2,2,3,1,8,'Водитель');
INSERT INTO
Employee(EmploeeID,Name,Emploee_Birth_date,TitleID,UnitID,DivisionID,ArmyID,DeptID,Emploee_specialisation) VALUES (12,'Медведев Д. А',TO_DATE ('03-03-1964', 'DD-MM-YYYY'),3,1,2,2,7,'Программист');

UPDATE Military_Unit
SET DislocationID = 1
WHERE UnitID = 3
UPDATE Military_Unit
SET DislocationID = 1
WHERE UnitID = 1
UPDATE Military_Unit
SET DislocationID = 2
WHERE UnitID = 2

```

На Рис. 2 изображен процесс создания базы данных в SQL Live ORACLE.

The screenshot shows the Oracle Live SQL interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (which is selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet". It contains a code editor with several SQL INSERT statements and their corresponding execution results. The results show 156 rows inserted into the Employee table, each with a unique EmployeeID and various names and birth dates. Below the code editor, there are several "1 row(s) inserted." messages. At the bottom of the page, there is footer information: "© 2020 Oracle Corporation - Privacy · Terms of Use", "Oracle Learning Library · Ask Tom · Dev Gym · Database Doc 19c , 18c , 12c · Follow on Twitter", "Live SQL 19.4.2, running Oracle Database 19c Enterprise Edition - 19.5.0.0.0", and "Built with ❤ using Oracle APEX".

**Рис. 2** Процесс создания БД в Live SQL Oracle

## Запросы в SQL Live ORACLE

1. Получить перечень всех частей военного округа, указанной армии, дивизии, корпуса
- a) Получим перечень всех военных частей (Military units) округа:

The screenshot shows the Oracle Live SQL interface. On the left is a code editor with the following SQL query:

```

1 select UnitID, Military_unit. Name
2 from Military_unit, Division
3 where Military_unit. DivisionID=Division.DivisionID;
4

```

To the right is a results grid titled "UNITID" and "NAME". It contains three rows of data:

UNITID	NAME
3	BBC
2	Танковая
1	Моторизованная

Below the grid, there are links for "Download CSV" and "3 rows selected."

- b) Получим перечень всех военных частей (Military units) указанной армии:

The screenshot shows the Oracle Live SQL interface. On the left is a code editor with the following SQL query:

```

1 SELECT
2 Military_Unit.UnitID, Military_Unit.Name
3 FROM
4 Military_Unit, Employee, Army
5 WHERE (Military_Unit.UnitID = Employee.UnitID and Employee.ArmyID = Army.ArmyID) and Army.ArmyID = 1;
6

```

To the right is a results grid titled "UNITID" and "NAME". It contains seven rows of data:

UNITID	NAME
1	Моторизованная
1	Моторизованная
3	BBC
1	Моторизованная
2	Танковая
2	Танковая
2	Танковая

Below the grid, there are links for "Download CSV" and "7 rows selected."

c) Получим перечень всех военных частей (Military units) дивизии:

```

1 SELECT
2 Military_Unit.UnitID, Military_Unit.Name
3 FROM Military_Unit, Division
4 WHERE
5 Military_Unit.DivisionID = Division.DivisionID and Division.DivisionID = 3;
6

```

UNITID	NAME
3	BBC

[Download CSV](#)

d) Получим перечень всех военных частей (Military units) корпуса (Construction):

```

1 SELECT Military_Unit.UnitID, Military_Unit.Name
2 FROM
3 Military_Unit, Construction
4 WHERE
5 Military_Unit.UnitID = Construction.UnitID and Construction.UnitID = 2;
6

```

UNITID	NAME
2	Танковая
2	Танковая

[Download CSV](#)

2 rows selected.

2. Получить данные по офицерскому составу в целом и по офицерскому составу указанного звания всех частей военного округа, отдельной армии, дивизии, корпуса, военной части:

a) Получим данные по офицерском составу в целом:

```

1 SELECT Employee.Name, Title.Name
2 FROM Title,Employee,Military_Unit
3 WHERE
4 Military_Unit.UnitID = Employee.UnitID and Employee.TitleID = Title.TitleID and Title.Title_Category = 'Офицер';
5

```

NAME	NAME
Гасанова З. А	Капитан
Иванова А. С	Майор
Волков Я. А	Лейтенант

[Download CSV](#)

3 rows selected.

b) Получим данные по офицерском составу указанного звания всех частей военного округа:

```

1 SELECT
2 Employee.Name
3 FROM
4 Title, Employee, Military_Unit
5 WHERE
6 Military_Unit.UnitID = Employee.UnitID and Employee.TitleID = Title.TitleID and Title.Name = 'Капитан';
7

```

NAME
Гасанова З. А

[Download CSV](#)

c) Получим данные по офицерском составу отдельной армии:

```

1 SELECT
2 Employee.Name, Employee.Name, Title.Name
3 FROM
4 Title, Employee, Army
5 WHERE
6 Army.ArmyID = Employee.ArmyID and Employee.TitleID = Title.TitleID and Title.Title_Category = 'Офицер' and Army.ArmyID = 2;
7
8

```

NAME	NAME	NAME
Волков Я. А	Волков Я. А	Лейтенант

d) Получим данные по офицерскому составу дивизии

NAME	NAME	NAME
Иванова А. С	Иванова А. С	Майор
Волков Я. А	Волков Я. А	Лейтенант

[Download CSV](#)  
2 rows selected.

e) Получим данные по офицерскому составу корпуса:

1 SELECT 2 Employee.Name, Title.Name 3 FROM 4 Title, Employee, Military_Unit, Construction 5 WHERE (Construction.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Employee.UnitID) 6 AND 7 Employee.TitleID = Title.TitleID and (Title.Title_Category = 'Офицер' and Construction.ConstructionID = 4); 8	NAME	NAME
	Иванова А. С	Майор

f) Получим данные по офицерскому составу военной части:

1 SELECT 2 Employee.Name, Title.Name 3 FROM 4 Title, Employee, Military_Unit 5 WHERE (Military_Unit.UnitID = Employee.UnitID and Employee.TitleID = Title.TitleID 6 and Title.Title_Category = 'Офицер' and Military_Unit.UnitID = 2)  7	NAME	NAME
	Волков Я. А	Лейтенант

3. Получить данные по рядовому и сержантскому составу в целом и с учетом указанного звания всех частей военного округа, отдельной армии, дивизии, корпуса, военной части:

a) Получим данные по рядовому и сержантскому составу в целом:

1 SELECT 2 Employee.Name, Title.Name 3 FROM 4 Title,Employee 5 WHERE 6 Title.TitleID = Employee.TitleID and (Title_Category = 'Сержантский состав' OR Title_Category = 'Рядовой состав'); 7	NAME	NAME
	Кормилицын А. Ю	Рядовой
	Узунова Е. Ф	Рядовой
	Бураки И. В	Старший сержант
	Петров В. И	Прапорщик
	Зубков А. Д	Сержант
	Маландин Г. И	Рядовой
	Герман А. А	Сержант
	Максаков И. П	Старший сержант
	Медведев Д. А	Ефрейтор

b) Получим данные по рядовому и сержантскому составу с учетом указанного звания всех частей военного округа:

1 SELECT 2 Employee.Name, Title.Name 3 FROM 4 Title, Employee, Military_Unit 5 WHERE 6 Military_Unit.UnitID = Employee.UnitID and Title.TitleID = Employee.TitleID and Title.Name = 'Рядовой'; 7   8	NAME	NAME
	Узунова Е. Ф	Рядовой
	Маландин Г. И	Рядовой

c) Получим данные по рядовому и сержантскому составу с учетом указанного звания отдельной армии:

d) Получим данные по рядовому и сержантскому составу с учетом указанного звания отдельной дивизии:

NAME	NAME
Кормилицын А. Ю	Рядовой
Узунова Е. Ф	Рядовой
Вураки И. В	Рядовой
Гасанова З. А	Рядовой
Иванова А. С	Рядовой
Петров В. И	Рядовой
Волков Я. А	Рядовой
Зубков А. Д	Рядовой
Мапандин Г. И	Рядовой
Герман А. А	Рядовой
Максаков И. П	Рядовой
Медведев Д. А	Рядовой
Кормилицын А. Ю	Рядовой
Узунова Е. Ф	Рядовой
Вураки И. В	Рядовой
Гасанова З. А	Рядовой
Иванова А. С	Рядовой
Петров В. И	Рядовой
Волков Я. А	Рядовой
Зубков А. Д	Рядовой

e) Получим данные по рядовому и сержантскому составу с учетом указанного звания отдельного корпуса:

NAME	NAME
Медведев Д. А	Ефрейтор
Вураки И. В	Старший сержант
Зубков А. Д	Сержант
Герман А. А	Сержант
Максаков И. П	Старший сержант
Петров В. И	Прапорщик
Медведев Д. А	Ефрейтор
Вураки И. В	Старший сержант
Зубков А. Д	Сержант
Герман А. А	Сержант
Максаков И. П	Старший сержант

[Download CSV](#)  
11 rows selected.

f) Получим данные по рядовому и сержантскому составу с учетом указанного

```

1 SELECT
2 Employee.Name, Title.Name
3 FROM
4 Title, Employee, Military_Unit
5 WHERE
6 Military_Unit.UnitID = Employee.UnitID AND Title.TitleID = Employee.TitleID
7 AND (Title_Category='Сержантский состав' OR Title_Category='Рядовой состав') AND Military_Unit.UnitID = 1;
8

```

NAME	NAME
Кормилицын А. Ю	Рядовой
Узунова Е. Ф	Рядовой
Медведев Д. А	Ефрейтор
Маландин Г. И	Рядовой

[Download CSV](#)

4 rows selected.

4. Получить цепочку подчиненности снизу доверху для указанного военнослужащего:

```

1 SELECT
2 Employee.Name, Title.Name
3 FROM
4 Title,Employee
5 WHERE
6 Title.TitleID = Employee.TitleID AND Title.reit > (SELECT Title.reit
7 FROM Title, Employee
8 WHERE Title.TitleID = Employee.TitleID AND Employee.Name = 'Кормилицын А. Ю')
9 ORDER BY Title.reit, Employee.Name, Title.Name asc;
10

```

NAME	NAME
Медведев Д. А	Ефрейтор
Герман А. А	Сержант
Зубков А. Д	Сержант
Вураки И. В	Старший сержант
Максаков И. П	Старший сержант
Петров В. И	Прапорщик
Волков Я. А	Лейтенант
Гасанова З. А	Капитан
Иванова А. С	Майор

[Download CSV](#)

9 rows selected.

5. Получить перечень мест дислокации всех частей военного округа, отдельной армии, дивизии, корпуса, военной части:

a) Получим перечень мест дислокации всех частей военного округа:

```

1 SELECT Dislocation_City
2 FROM Dislocation, Military_Unit
3 WHERE Military_Unit.DislocationID = Dislocation.DislocationID;
4

```

DISLOCATION\_CITY

Краснодар

Уфа

Краснодар

[Download CSV](#)

3 rows selected.

b) Получим перечень мест дислокации всех частей отдельной армии:

```

1 SELECT Dislocation_City
2 FROM
3 Dislocation, Military_Unit, Employee, Army
4 WHERE Military_Unit.UnitID = Dislocation.DislocationID AND Military_Unit.UnitID = Employee.UnitID
5 AND Employee.ArmyID = Army.ArmyID AND Army.ArmyID = 1;
6

```

DISLOCATION\_CITY

Краснодар

Краснодар

Краснодар

Уфа

Уфа

Уфа

Краснодон

[Download CSV](#)

7 rows selected.

с) Получим перечень мест дислокации(Dislocation\_City) всех частей дивизии:

```
1 SELECT Dislocation_City
2 FROM Dislocation, Military_Unit, Division
3 WHERE Military_Unit.DislocationID = Dislocation.DislocationID and Military_Unit.DivisionID = Division.DivisionID
4 and Division.DivisionID = 2;
```

DISLOCATION_CITY
Уфа

[Download CSV](#)

д) Получим перечень мест дислокации (Dislocation\_City) всех частей корпуса:

```
1 SELECT Dislocation_City
2 FROM Dislocation, Construction
3 WHERE Construction.ConstructionID = Dislocation.ConstructionID and Construction.ConstructionID = 1;
4
```

DISLOCATION_CITY
Краснодар
Севастополь
Новоросийск

[Download CSV](#)

3 rows selected.

е) Получим перечень мест дислокации(Dislocation\_City) всех частей военной части:

```
1 SELECT Dislocation_City
2 FROM Dislocation, Military_Unit
3 WHERE Military_Unit.DislocationID = Dislocation.DislocationID and Military_Unit.UnitID = 2;
4
```

DISLOCATION_CITY
Уфа

[Download CSV](#)

6. Получить данные о наличии боевой технике в целом и с учетом указанной категории или вида во всех частях военного округа, в отдельной армии, дивизии, корпусе, военной части:

а) Получим данные о наличии боевой техники(Weapon) в целом во всех частях военного округа:

```
1 SELECT Tehnician.Name
2 FROM Tehnician, Military_Unit
3 WHERE Military_Unit.UnitID = Tehnician.UnitID;
4
5
```

NAME
ДДТ
Грузовик
Танк Т-34
Грузовик
Танк Т-34
Платон
Танк Т-62
Грузовик
Козак
Козак-М2
Козак-М2
Козак-М1
Козак
Оплот

[Download CSV](#)

14 rows selected.

b) Получим данные о наличии боевой техники(Weapon) в отдельной армии:

```
1 SELECT Tehnician.Name, Tehnician.TehnicianID
2 FROM Tehnician, Military_Unit, Emploee, Army
3 WHERE Tehnician.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Emploee.UnitID and
4 Emploee.ArmyID = Army.ArmyID and Army.ArmyID = 3;
5
```

NAME	TEHNICIANID
ДДТ	1
Танк Т-34	3
Грузовик	4
Танк Т-34	5
Грузовик	8
Козак	9
Козак-М2	10
Козак-М2	11
Козак-М1	12
Козак	13
Оплот	14

[Download CSV](#)

11 rows selected.

c) Получим данные о наличии боевой техники(Weapon) в дивизии:

```
1 SELECT Tehnician.Name, Tehnician.TehnicianID
2 FROM Tehnician, Military_Unit, Division
3 WHERE Tehnician.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Division.DivisionID
4 and Division.DivisionID = 2;
5
6
```

NAME	TEHNICIANID
Грузовик	2
Платон	6
Танк Т-62	7

[Download CSV](#)

3 rows selected.

d) Получим данные о наличии боевой техники(Weapon) в корпусе:

```
1 SELECT Tehnician.Name, Tehnician.TehnicianID
2 FROM Tehnician, Military_Unit, Construction
3 WHERE Tehnician.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Construction.UnitID
4 and Construction.ConstructionID = 3;
5
```

NAME	TEHNICIANID
Танк Т-34	3
Грузовик	8
Козак	9

[Download CSV](#)

3 rows selected.

e) Получим данные о наличии боевой техники(Weapon) в военной части:

```

1 SELECT Tehnician.Name
2 FROM Tehnician, Military_Unit
3 WHERE Tehnician.UnitID = Tehnician.UnitID and Military_Unit.UnitID = 2;
4

```

NAME
ДДТ
Грузовик
Танк Т-34
Грузовик
Танк Т-34
Платон
Танк Т-62
Грузовик
Козак
Козак-М2
Козак-М2
Козак-М1
Козак
Оплот

Download CSV  
14 rows selected.

7. Получить перечень сооружений указанной военной части, перечень сооружений, где дислоцировано более одного подразделения, где не дислоцировано ни одного подразделения:
- a) Получим перечень сооружений указанной военной части:
8. Получить перечень военных частей, в которых число единиц указанного вида боевой техники больше 5 (нет указанной боевой техники):

```

1 SELECT Military_Unit.UnitID AS Military_Unit, COUNT (Tehnician.TehnicianID) AS Tehnician
2 FROM Military_Unit INNER JOIN Tehnician ON Tehnician.UnitID = Military_Unit.UnitID
3 WHERE Tehnician.Name = 'БМП'
4 GROUP BY Military_Unit.UnitID
5 HAVING COUNT (Tehnician.TehnicianID) > 5;
6

```

no data found

9. Получить данные о наличии вооружения в целом и с учетом указанной категории или вида во всех частях военного округа, в отдельной армии, дивизии, корпусе, военной части:
- a) Получим данные о наличии вооружения в целом во всех частях военного округа:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit
3 WHERE
4 Military_Unit.UnitID = Weapon.UnitID;
5

```

NAME
Ракетное вооружение
Артилерия
Автоматическое оружие
Ракетное вооружение
Артилерия
Автоматическое оружие

Download CSV  
6 rows selected.

b) Получим данные о наличии вооружения в целом в отдельной армии:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Employe, Army
3 WHERE Weapon.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Employe.UnitID and
4 Employe.ArmyID = Army.ArmyID and Army.ArmyID = 3;
5

```

NAME
Ракетное вооружение
Автоматическое оружие
Ракетное вооружение
Артилерия

Download CSV  
4 rows selected.

c) Получим данные о наличии вооружения в целом в дивизии:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Division
3 WHERE Weapon.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Division.DivisionID and
4 Division.DivisionID = 3;

```

NAME
Автоматическое оружие
Ракетное вооружение

[Download CSV](#)  
2 rows selected.

d) Получим данные о наличии вооружения в целом в корпусе:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Construction
3 WHERE Weapon.UnitID = Military_Unit.UnitID and Military_Unit.UnitID = Construction.UnitID and
4 Construction.ConstructionID = 2;
5

```

NAME
Артилерия
Автоматическое оружие

[Download CSV](#)  
2 rows selected.

e) Получим данные о наличии вооружения в целом в военной части:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit
3 WHERE Weapon.UnitID = Weapon.UnitID and Military_Unit.UnitID = 1;
4
5

```

NAME
Ракетное вооружение
Артилерия
Автоматическое оружие
Ракетное вооружение
Артилерия
Автоматическое оружие

[Download CSV](#)  
6 rows selected.

f) Получим данные о наличии вооружения с учетом указанной категории в военной части:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Tehnician
3 WHERE Military_Unit.UnitID = Weapon.UnitID and Military_Unit.UnitID = Tehnician.UnitID
4 and Tehnician.Name = 'Козак' and Military_Unit.UnitID = 3;
5

```

NAME
Автоматическое оружие
Ракетное вооружение

[Download CSV](#)  
2 rows selected.

g) Получим данные о наличии вооружения с учетом указанной категории в отдельной армии:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Tehnician, Employee, Army
3 WHERE Military_Unit.UnitID = Weapon.UnitID and Military_Unit.UnitID = Tehnician.UnitID and Military_Unit.UnitID = Employee.UnitID
4 and Tehnician.Name='Козак-M2' and Employee.ArmyID = Army.ArmyID and Army.ArmyID=3;
5

```

NAME
Ракетное вооружение
Ракетное вооружение
Артилерия
Артилерия

[Download CSV](#)  
4 rows selected.

h) Получим данные о наличии вооружения с учетом указанной категории в дивизии:

```

1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Tehnician, Division
3 WHERE Military_Unit.UnitID = Weapon.UnitID and Military_Unit.UnitID = Tehnician.UnitID and Military_Unit.UnitID = Division.DivisionID
4 and Tehnician.Name='Козак-M2' and Division.DivisionID = 1;
5

```

NAME
Ракетное вооружение
Артилерия
Ракетное вооружение
Артилерия

[Download CSV](#)  
4 rows selected.

i) Получим данные о наличии вооружения с учетом указанной категории в корпусе:

```
1 SELECT Weapon.Name
2 FROM Weapon, Military_Unit, Tehnician, Construction
3 WHERE Military_Unit.UnitID = Weapon.UnitID and Military_Unit.UnitID = Tehnician.UnitID and Military_Unit.UnitID = Construction.UnitID
4 and Tehnician.Name='Танк Т-34' and Construction.ConstructionID = 3;
5
```

NAME
Автоматическое оружие
Ракетное вооружение

[Download CSV](#)

2 rows selected.

10. Получить перечень военных специальностей, по которым в округе, в отдельной армии, дивизии, корпусе, военной части более пяти специалистов (нет специалистов):

```
1 SELECT Military_Unit.UnitID AS Military_Unit, COUNT (Employee.EmployeeID) AS Employee
2 FROM Employee INNER JOIN Military_Unit ON Military_Unit.UnitID = Employee.UnitID
3 WHERE Employee.Employee_specialisation = 'Инженер'
4 GROUP BY Military_Unit.UnitID
5 HAVING COUNT (Employee.EmployeeID) > 5;
6
```

no data found

11. Получить перечень военнослужащих указанной специальности в округе, в отдельной армии, дивизии, корпусе, военной части, в указанном подразделении некоторой военной части:

a) Получим перечень военнослужащих указанной специальности в округе:

```
1 SELECT Employee.Name
2 FROM Employee, Military_Unit
3 WHERE Employee.UnitID = Military_Unit.UnitID and Employee.Employee_Specialisation = 'Программист';
4
```

NAME
Медведев Д. А

[Download CSV](#)

b) Получим перечень военнослужащих указанной специальности в дивизии:

```
1 SELECT Employee.Name
2 FROM Employee, Division
3 WHERE Employee.DivisionID = Division.DivisionID and Employee.Employee_Specialisation = 'Инженер';
4
```

NAME
Вураки И. В
Петров В. И
Манандин Г. И

[Download CSV](#)

3 rows selected.

12. Получить перечень военных частей, в которых число единиц указанного вида вооружения больше 10 (нет указанного вооружения).

```
1
2 SELECT Military_Unit.UnitID AS Military_Unit, COUNT (Weapon.WeaponID) AS Weapon
3 FROM Weapon INNER JOIN Military_Unit ON Military_Unit. UnitID = Weapon.UnitID
4 WHERE Weapon.Name = 'Артилерия'
5 GROUP BY Military_Unit.UnitID
6 HAVING COUNT (Weapon.WeaponID) > 10;
7
```

no data found

13. Получить данные об армии, дивизии, корпусе, в которые входит больше всего (меньше всего) военных частей.

a) Больше всего:

```
1 SELECT Division.Numb
2 from Division, Military_Unit
3 where Division.DivisionID = Military_Unit.DivisionID
4 and Military_Unit.UnitID >= (select distinct count (*) from Military_Unit);
5
```

NUMB

3

[Download CSV](#)

b) Меньше всего:

```
1 SELECT Division.Numb
2 from Division, Military_Unit
3 where Division.DivisionID = Military_Unit.DivisionID
4 and Military_Unit.UnitID < (select distinct count (*) from Military_Unit);
5
```

NUMB

1

2

[Download CSV](#)

2 rows selected.

## **Выводы**

В ходе решения данной курсовой работы я подробно ознакомился с теорией и практикой создания комплексных информационных систем, в контексте тематики данной работы – информационной системы военного округа. Была изучена предметная область задачи, в ходе изучения предметной области было предложено использовать ряд сущностей (Entities) для создания реляционной модели данных (Relational – Entities model).

Построена ER-модель военного округа в среде draw.io, которая позволила упростить процесс проектирования и документирования базы данных, ER-модель на Рис.1 помогла визуализировать структуру данных, одновременно обеспечивая эффективный процесс создания и администрирования базы данных военного округа.

SQL Live ORACLE является инструментом для разработки и администрирования баз данных в режиме онлайн, благодаря браузерному интерфейсу, SQL Live ORACLE – это кроссплатформенный инструмент для работы с SQL – запросами, что является несомненным преимуществом данной программы.

В среде SQL Live ORACLE была создана база данных военного округа и произведены все SQL-запросы, указанные в задании к курсовой работе. Все запросы подробно задокументированы и указаны в работе.

В ходе написании курсовой работы были получены первоначальные навыки в разработке и развёртывании мощных информационных приложений.

## **Литература**

- [1] **Rames Elmasri, Shamkant B. Navathe**, Fundamentals of Database systems 7<sup>th</sup> edition, Dep. Of computer Science and Engineering, The University of Texas at Arlington, 2016.
- [2] Дейт Кристофер, Введение в системы баз данных. – М.: Вильямс, 2006.
- [3] Online – course: «Databases» :  
URL : <https://www.coursera.org/learn/data-bases-intr/home/info>
- [4] Электронная библиотека SQL. – URL : <http://www.sql-tutorial.ru>
- [5] Live SQL ORACLE. [Электронный ресурс]. – URL : <https://livesql.oracle.com>

## Приложения

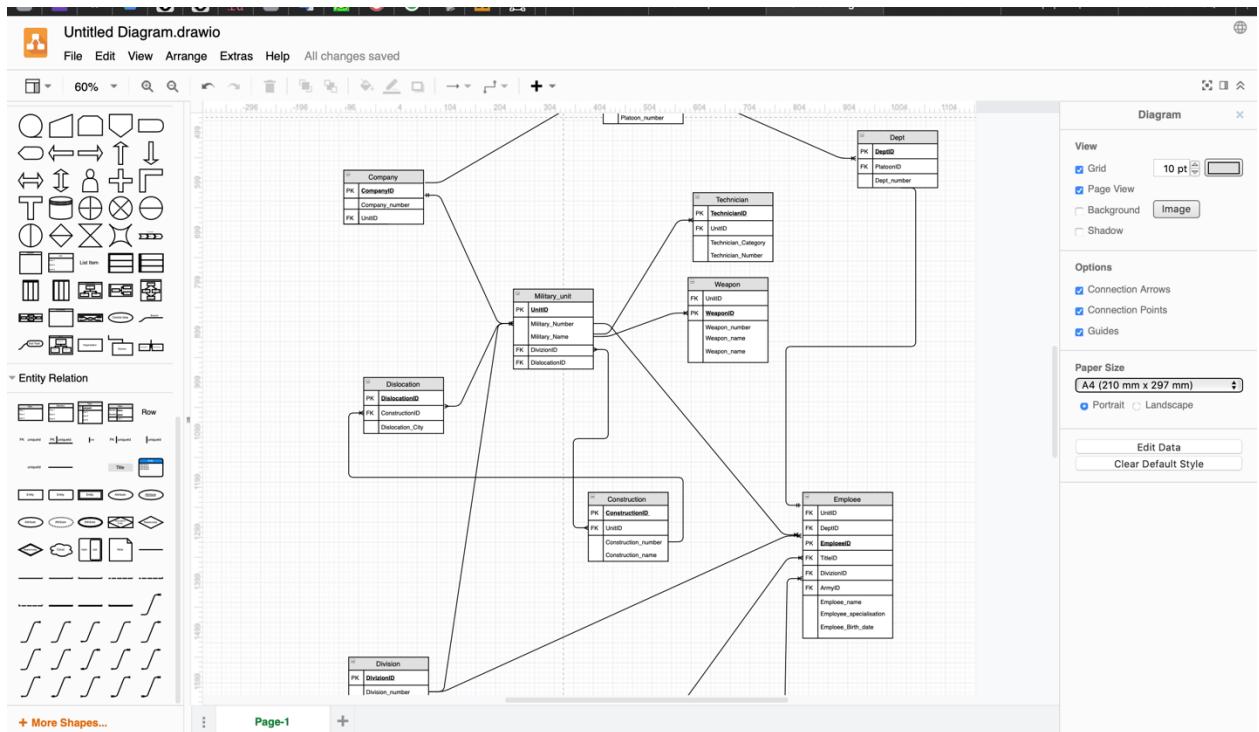


Рис. 3 Интерфейс в программе draw.io

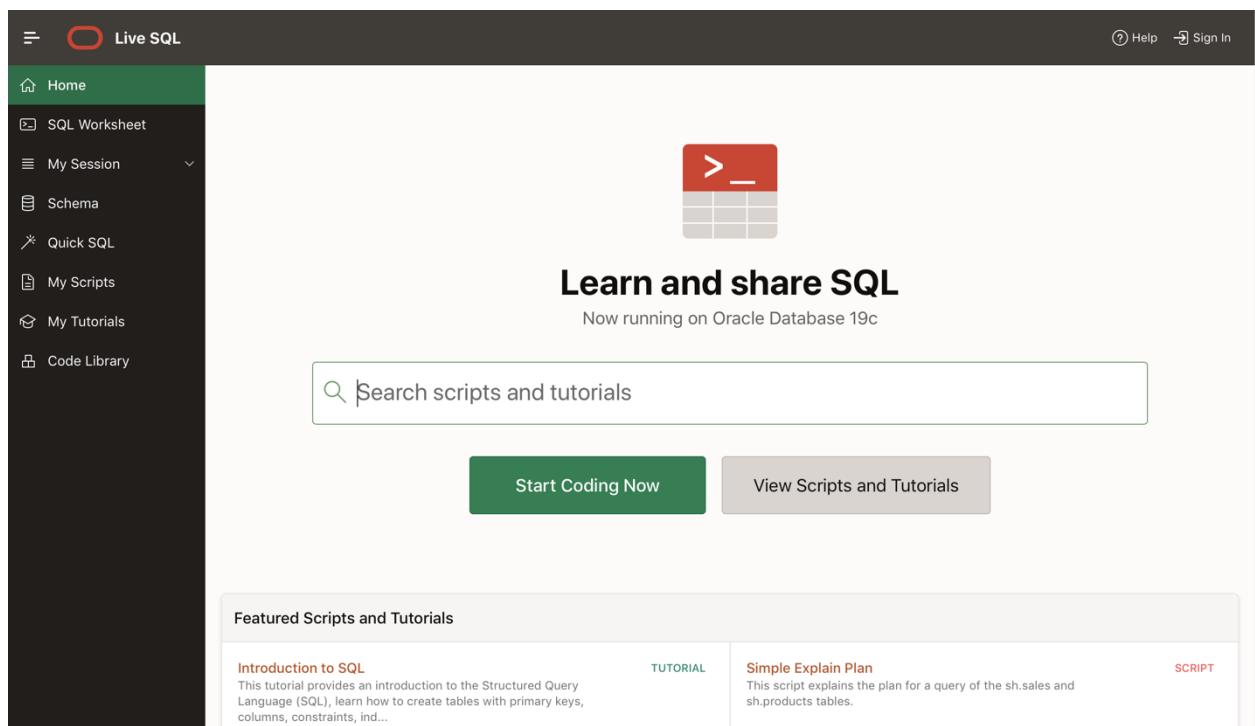


Рис. 4 Интерфейс в программе live SQL ORACLE