**Отчет по практике**

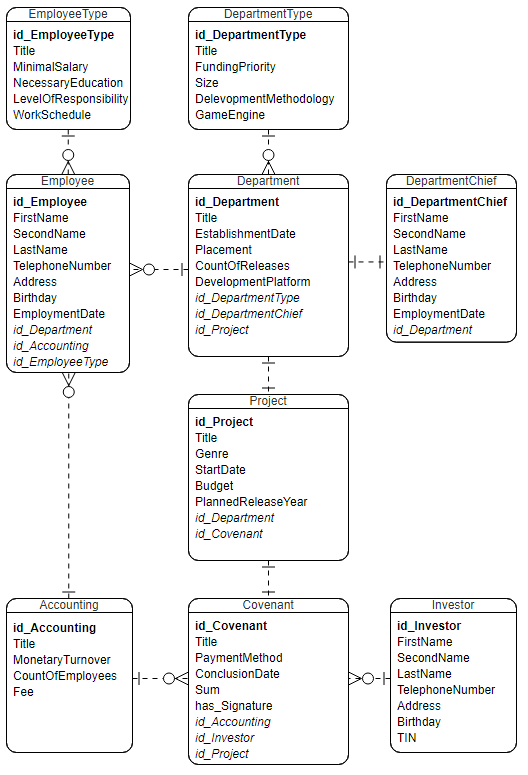
**студента группы И-31**

**Сафронова Данила**

**Содержание**

1. ER-диаграмма
2. Схема данных
3. Описание таблиц и атрибутов
4. Нормализация
5. SQL-запросы на создание таблиц
6. SQL-запрос на создание индексов
7. SQL-запросы на вывод данных
8. Резервное копирование
9. Инструкция для пользователя и для администратора

1. ER-диаграмма



**2. Схема данных**



**3. Описание таблиц и атрибутов**

1) Таблица EmployeeType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_EmployeeType | int | 10 | NOT NULL | + |  | Код типа сотрудника |
| Title | varchar | 50 |  |  |  | Название |
| MinimalSalary | money | 15 |  |  |  | Минимальная зарплата для данного типа сотрудников |
| NecessaryEducation | varchar | 20 |  |  |  | Необходимое образование |
| LevelOfResponsibility | varchar | 20 |  |  |  | Уровень ответственности |
| WorkSchedule | varchar | 20 |  |  |  | График работы |

2) Таблица Employee

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Employee | int | 10 | NOT NULL | + |  | Код типа сотрудника |
| FirstName | varchar | 50 |  |  |  | Имя |
| SecondName | varchar | 50 |  |  |  | Отчество |
| LastName | varchar | 50 |  |  |  | Фамилия |
| TelephoneNumber | varchar | 11 |  |  |  | Номер телефона |
| Address | varchar | 30 |  |  |  | Адрес |
| Birthday | date | 10 |  |  |  | Дата рождения |
| EmploymentDate | date | 10 |  |  |  | Дата трудоустройства |
| id\_Department | int | 10 | NOT NULL |  | + | Код отдела |
| id\_Accounting | int | 10 | NOT NULL |  | + | Код бухгалтерии |
| id\_EmployeeType | int | 10 | NOT NULL |  | + | Код типа сотрудника |

3) Таблица Department

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Deployment | int | 10 | NOT NULL | + |  | Код отдела |
| Title | varchar | 50 |  |  |  | Название |
| EstablishmentDate | date | 10 |  |  |  | Дата учреждения |
| Placement | varchar | 20 |  |  |  | Местонахождение |
| CountOfReleases | int | 3 |  |  |  | Количество выпущенных игр |
| DevelopmentPlatform | varchar | 20 |  |  |  | Платформа, для которой разрабатываются игры (Windows, xbox, playstation) |
| id\_DepartmentType | int | 10 | NOT NULL |  | + | Код типа отдела |
| id\_DepartmentChief | int | 10 | NOT NULL |  | + | Код руководителя отдела |

4) Таблица DepartmentType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_DeploymentType | int | 10 | NOT NULL | + |  | Код типа отдела |
| Title | varchar | 50 |  |  |  | Название |
| FundingPriority | varchar | 15 |  |  |  | Приоритет финансирования |
| Size | int | 10 |  |  |  | Размер (количество сотрудников) |
| DevelopmentMethodology | varchar | 20 |  |  |  | Методология разработки |
| GameEngine | varchar | 20 |  |  |  | Игровой движок |

5) Таблица DepartmentChief

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_DepartmentChief | int | 10 | NOT NULL | + |  | Код руководителя отдела |
| FirstName | varchar | 50 |  |  |  | Имя |
| SecondName | varchar | 50 |  |  |  | Отчество |
| LastName | varchar | 50 |  |  |  | Фамилия |
| TelephoneNumber | varchar | 20 |  |  |  | Номер телефона |
| Address | varchar | 20 |  |  |  | Адрес |
| Birthday | date | 10 |  |  |  | Дата рождения |
| EmploymentDate | date | 10 |  |  |  | Дата трудоустройства |
| id\_Department | int | 10 |  |  | + | Код отдела |

6) Таблица Project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Project | int | 10 | NOT NULL | + |  | Код проекта |
| Title | varchar | 50 |  |  |  | Название |
| Genre | varchar | 15 |  |  |  | Жанр |
| StartDate | date | 10 |  |  |  | Дата начала разработки |
| Budget | money | 15 |  |  |  | Бюджет |
| PlannedReleaseYear | int | 4 |  |  |  | Планируемый год выпуска |
| id\_Department | int | 10 | NOT NULL |  | + | Код отдела |
| id\_Covenant | int | 10 | NOT NULL |  | + | Код договора |

7) Таблица Covenant

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Covenant | int | 10 | NOT NULL | + |  | Код договора |
| Title | varchar | 50 |  |  |  | Название |
| PaymentMethod | varchar | 15 |  |  |  | Способ оплаты |
| ConclusionDate | date | 10 |  |  |  | Дата заключения |
| Sum | money | 10 |  |  |  | Сумма договора |
| has\_Signature | varchar | 20 |  |  |  | Имеет ли подпись |
| id\_Accounting | int | 10 | NOT NULL |  | + | Код бухгалтерии |
| id\_Investor | int | 10 | NOT NULL |  | + | Код инвестора |

8) Таблица Investor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Investor | int | 10 | NOT NULL | + |  | Код инвестора |
| FirstName | varchar | 50 |  |  |  | Имя |
| SecondName | varchar | 50 |  |  |  | Отчество |
| LastName | varchar | 50 |  |  |  | Фамилия |
| TelephoneNumber | int | 10 |  |  |  | Номер телефона |
| Address | varchar | 20 |  |  |  | Адрес |
| Birthday | date | 10 |  |  |  | Дата рождения |
| TIN | int | 10 |  |  |  | Номер ИНН |

9) Таблица Accounting

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | Длина | Допустимое значение | Первичный ключ | Внешний ключ | Описание |
| id\_Accounting | int | 10 | NOT NULL | + |  | Код бухгалтерии |
| Title | varchar | 50 |  |  |  | Название |
| MonetaryTurnoverPerYear | money | 15 |  |  |  | Годовой денежный оборот |
| CountOfEmployees | int | 10 |  |  |  | Количество сотрудников |
| Fee | money | 10 |  |  |  | Комиссия за заключение договоров |
| id\_Employee | varchar | 20 | NOT NULL |  | + | Код сотрудника |

**4. Нормализация**

EmployeeType (**id\_EmployeeType**, Title, MinimalSalary, NecessaryEducation, LevelOfResponsibility, WorkSchedule)

Employee (**id\_Employee**, FirstName, SecondName, LastName, TelephoneNumber, Address, Birthday, EmploymentDate, id\_Department, id\_Accounting)

Department (**id\_Department**, Title, EstablishmentDate, Placement, CountOfReleases, DevelopmentPlatform, id\_DepartmentType, id\_DepartmentChief)

DepartmentType (**id\_DepartmentType**, Title, FundingPriority, Size, DevelopmentMethodology, GameEngine)

DepartmentChief (**id\_DepartmentChief**, FirstName, SecondName, LastName, TelephoneNumber, Address, Birthday, EmploymentDate, id\_Department)

Project (**id\_Project**, Title, Genre, StartDate, Budget, PlannedReleaseYear, id\_Department, id\_Covenant)

Covenant (**id\_Covenant**, Title, PaymentMethod, ConclusionDate, Sum, has\_Signature, id\_Accounting, id\_Investor)

Investor (**id\_Investor**, FirstName, SecondName, LastName, TelephoneNumber, Address, Birthday, TIN)

Accounting (**id\_Accounting**, Title, MonetaryTurnover, CountOfEmployees, Fee, id\_Employee)

**5. SQL-запросы на создание таблиц**

Код запроса:

create table EmployeeType (

id\_EmployeeType int not null,

Title varchar(50) null,

MinimalSalary money null,

NecessaryEducation varchar(20) null,

LevelOfResponsibility varchar(20) null,

WorkSchedule varchar(20) null,

);

create table Employee(

id\_Employee int not null,

FirstName varchar(50) null,

SecondName varchar(50) null,

LastName varchar(50) null,

TelephoneNumber varchar(20) null,

Address varchar(30) null,

Birthday date null,

EmploymentDate date null,

id\_Department int not null,

id\_Accounting int not null,

id\_EmployeeType int not null

);

create table DepartmentType(

id\_DepartmentType int not null,

Title varchar(50) null,

FundingPriority varchar(15) null,

Size int null,

DevelopmentMethodology varchar(20) null,

GameEngine varchar(20) null

);

create table Department(

id\_Department int not null,

Title varchar(50) null,

EstablishmentDate date null,

Placement varchar(20) null,

CountOfReleases int null,

DevelopmentPlatform varchar(20) null,

id\_DepartmentType int not null,

id\_DepartemntChief int not null,

id\_Project int not null

);

create table DepartmentChief(

id\_DepartmentChief int not null,

FirstName varchar(50) null,

Secondname varchar(50) null,

LastName varchar(50) null,

TelephoneNumber varchar(11) null,

Address varchar(20) null,

Birthday date null,

EmploymentDate date null,

id\_Department int not null

);

create table Project(

id\_Project int not null,

Title varchar(50) null,

Genre varchar(15) null,

StartDate date null,

Budget money null,

PlannedReleaseYear int null,

id\_Department int not null,

id\_Covenant int not null

);

create table Covenant(

id\_Covenant int not null,

Title varchar(50) null,

PaymentMethod varchar(15) null,

ConclusionDate date null,

Sum money null,

has\_Signature varchar(20) null,

id\_Accounting int not null,

id\_Investor int not null,

id\_Project int not null

);

create table Accounting(

id\_Accounting int not null,

Title varchar(50) null,

MonetaryTurnoverPerYear money null,

CountOfEmployees int null,

Fee varchar(3) null,

id\_Employee int not null

);

create table Investor(

id\_Investor int not null,

FirstName varchar(50) null,

SecondName varchar(50) null,

LastName varchar(50) null,

TelephoneNumber varchar(20) null,

Address varchar(20) null,

Birthday date null,

TIN varchar(20) null

);

**6. SQL-запрос на создание индексов**

Код запроса:

create index idx\_MonetaryTurnover

on Accounting(MonetaryTurnoverPerYear)

create index idx\_AccountingID

on Accounting(id\_Accounting)

create index idx\_CovenantID

on Covenant(id\_Covenant)

create index idx\_PaymentMethodCovenant

on Covenant(PaymentMethod)

create index idx\_DepartmentID

on Department(id\_Department)

create index idx\_DevelopmentPlatform

on Department(DevelopmentPlatform)

create index idx\_FirstNameDepartmentChief

on DepartmentChief(FirstName)

create index idx\_SecondNameDepartmentChief

on DepartmentChief(SecondName)

create index idx\_DepartmentTypeID

on DepartmentType(id\_DepartmentType)

create index idx\_DepartmentTypeFundingPriority

on DepartmentType(FundingPriority)

create index idx\_FirstNameEmployee

on Employee(FirstName)

create index idx\_SecondNameEmployee

on Employee(SecondName)

create index idx\_TelephoneNumberEmployee

on Employee(TelephoneNumber)

create index idx\_EmployeeID

on Employee(id\_Employee)

create index idx\_TelephoneNumberEmployee

on Employee(TelephoneNumber)

create index idx\_EmployeeID

on Employee(id\_Employee)

create index idx\_EmployeeTypeID

on EmployeeType(id\_EmployeeType)

create index idx\_EmployeeTypeLevelOfResponsibility

on EmployeeType(LevelOfResponsibility)

create index idx\_FirstNameInvestor

on Investor(FirstName)

create index idx\_SecondNameInvestor

on Investor(SecondName)

create index idx\_Project\_DepartmentID

on Project(id\_Department)

create index idx\_ProjectGenre

on Project(Genre)

**7. SQL-запросы на вывод данных**

1) Вывести фамилию и имя всех руководителей, ответственных за проекты, разработка которых началась в 2023 году, имя которых начинается на букву M.

Код запроса:

select DepartmentChief.FirstName, DepartmentChief.SecondName

from DepartmentChief

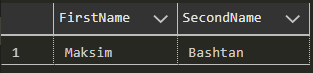
join Department on DepartmentChief.id\_DepartmentChief = Department.id\_DepartmentChief

join Project on Department.id\_Department = Project.id\_Department

where year(Project.StartDate) = 2023

and DepartmentChief.FirstName like 'M%'

Результат запроса:



2) Вывести количество договоров с отсутствующей подписью, их номера, а также фамилию и инициалы инвестора, сгруппированные по инвестору, который этот договор хочет заключить. Вывести 10 результатов, используя функцию top. Отсортировать по номеру договора.

Код запроса:

select top 10

count(Covenant.id\_Covenant) as count,

Covenant.id\_Covenant as CovenantID,

Investor.SecondName as SecondName,

SUBSTRING(Investor.FirstName, 1, 1) as FirstName,

SUBSTRING(Investor.LastName, 1, 1) as LastName

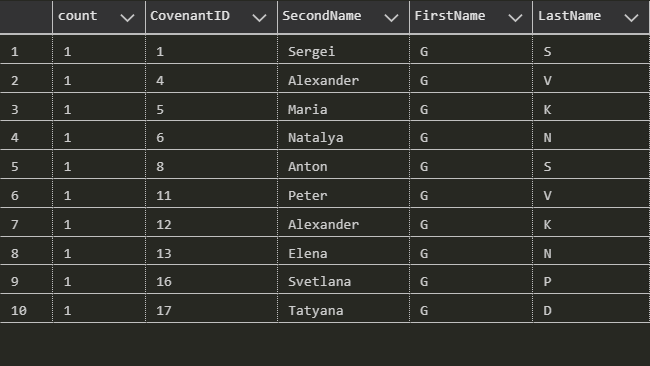
from Covenant join Investor on Covenant.id\_Investor = Investor.id\_Investor

where Covenant.has\_Signature = 'yes'

group by SecondName, FirstName, LastName, Covenant.id\_Covenant

order by Covenant.id\_Covenant

Результат запроса:



3) Вывести ФИО всех сотрудников и разрабатываемые их отделом, применяющим методологию разработки «Scrum», проекты

Код запроса:

select Employee.FirstName, Employee.SecondName, Employee.LastName, Project.Title

from Employee

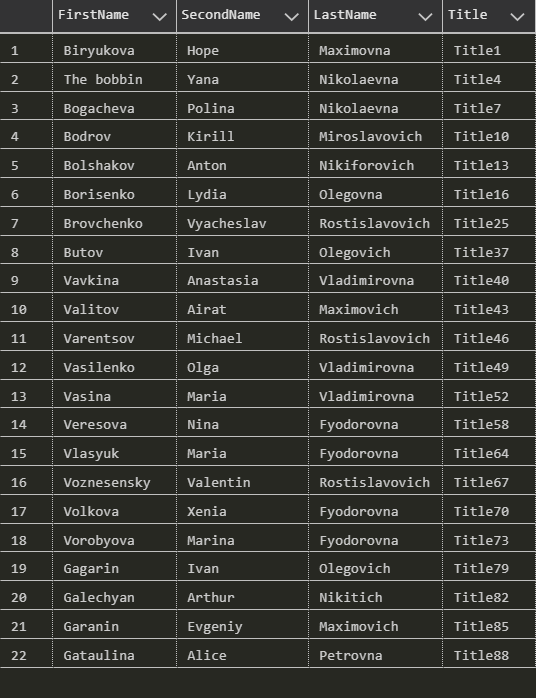
join Department on Employee.id\_Department = Department.id\_Department

join DepartmentType on DepartmentType.id\_DepartmentType = Department.id\_DepartmentType

join Project on Project.id\_Department = Department.id\_Department

where DepartmentType.DevelopmentMethodology = 'Scrum'

Результат запроса:



4) Вывести имя и номер самого молодого сотрудника, работающего в отделе, руководитель которого родился в 2000 году

Код запроса:

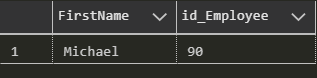
select Employee.FirstName, Employee.id\_Employee from Employee

join Department on Department.id\_Department = Employee.id\_Department

join DepartmentChief on DepartmentChief.id\_DepartmentChief = Department.id\_DepartmentChief

where year(DepartmentChief.Birthday) = 2000 and Employee.Birthday = (select max(birthday) from Employee)

Результат запроса:



5) Вывести 10 записей, включающие номер бухгалтерии и ее средний денежный оборот за год, если она заключила хотя бы один договор, оплаченный наличными. Отсортировать по обороту за год в порядке возрастания.

Код запроса:

select top 10

Accounting.id\_Accounting,

avg(Accounting.MonetaryTurnoverPerYear) as AvgTurnover

from Accounting

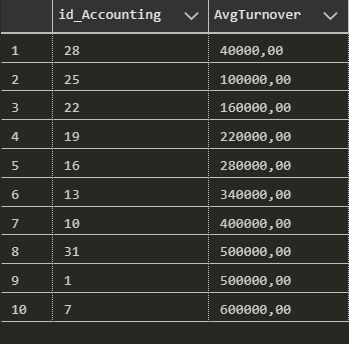
join Covenant on Covenant.id\_Accounting = Accounting.id\_Accounting

where Covenant.PaymentMethod = 'Cash'

group by Accounting.id\_Accounting

order by AvgTurnover

Результат запроса:



6) Создать представление, в котором будут показаны ФИО сотрудника и его номер телефона в формате «8\*\*\*99». Например, 89999999912 будет равен 8\*\*\*12. Вывести первые 10 записей этого представления.

Код запроса для создания представления:

create view EmployeeView as

select FirstName, SecondName, LastName,

concat(substring(TelephoneNumber, 1, 1), '\*\*\*', SUBSTRING(TelephoneNumber, 10, 11)) as TelephoneNumber

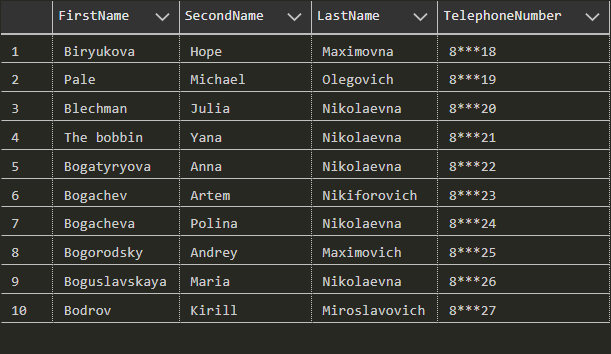
from Employee

Код запроса для вывода информации из созданного представления:

select top 10 \*

from EmployeeView

Результат запроса:



7) Вывести имя, фамилию и название месяца, в котором родились руководители отделов, ответственные за разработку проектов в жанре «Souls-like»

Код запроса:

select FirstName, LastName,

case

when month(Birthday) = 1 then 'Born in January'

when month(Birthday) = 2 then 'Born in February'

when month(Birthday) = 3 then 'Born in March'

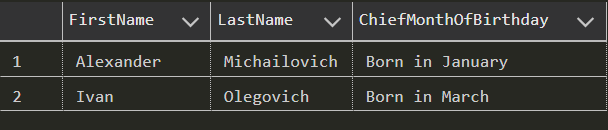
end as ChiefMonthOfBirthday

from DepartmentChief

join Department on Department.id\_DepartmentChief = DepartmentChief.id\_DepartmentChief

join Project on Project.id\_Department = Department.id\_Project

where Project.Genre = 'Soulslike'

Результат запроса:

8) Вывести сумму бюджета всех проектов, разрабатываемых отделами с высоким приоритетом финансирования, в которых работают сотрудники с высоким уровнем ответственности. Вынести логику выражений join в отдельный блок с конструкцией with.

Код запроса:

with HighestPriority(Budget) as

(select Budget from Project

join Department on Project.id\_Department = Department.id\_Project

join DepartmentType on Department.id\_DepartmentType = DepartmentType.id\_DepartmentType

join Employee on Employee.id\_Department = Department.id\_Department

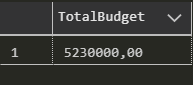
join EmployeeType on EmployeeType.id\_EmployeeType = Employee.id\_EmployeeType

where DepartmentType.FundingPriority = 'High' and EmployeeType.LevelOfResponsibility = 'High'

)

select sum(Budget) as TotalBudget from HighestPriority

Результат запроса:



9) Сравнить для какой платформы (Windows, Xbox, Playstation) в данный момент создается наибольшее число проектов на движке Unity. В результате запроса вывести название платформы.

Код запроса:

declare @WindowsCount int = (select count(id\_Department) from Department where DevelopmentPlatform = 'Windows')

declare @XboxCount int = (select count(id\_Department) from Department where DevelopmentPlatform = 'Xbox')

declare @PlaystationCount int = (select count(id\_Department) from Department where DevelopmentPlatform = 'Playstation')

select distinct case

when

@WindowsCount > @PlaystationCount and @WindowsCount > @XboxCount

then 'Windows'

when

@XboxCount > @PlaystationCount and @XboxCount > @WindowsCount

then 'Xbox'

when

@PlaystationCount > @WindowsCount and @PlaystationCount > @XboxCount

then 'Playstation'

end

as MostPopularPlatform

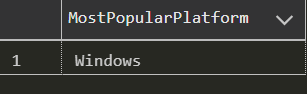
from Department

join DepartmentType

on DepartmentType.id\_DepartmentType = Department.id\_DepartmentType

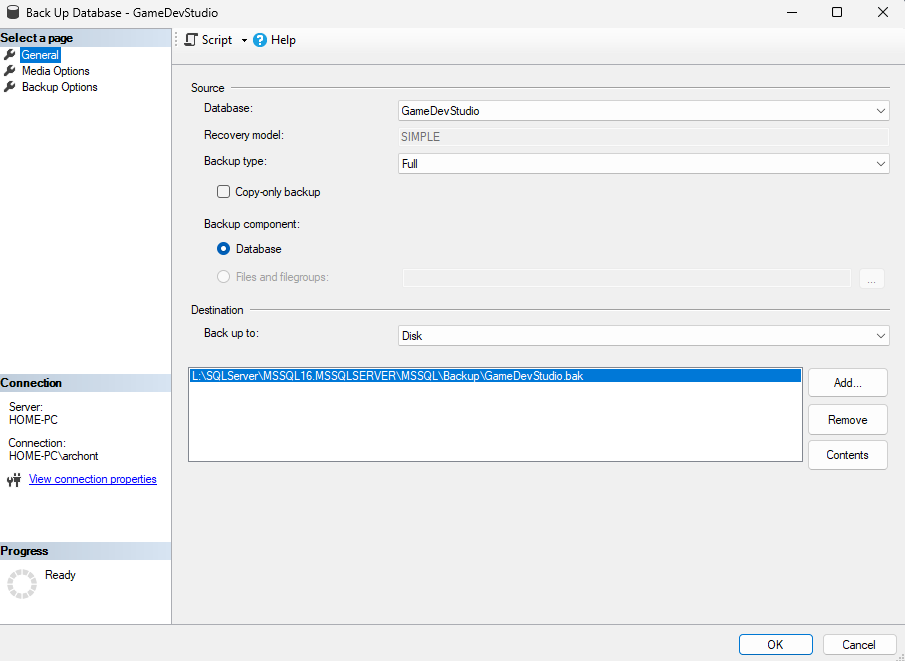
where DepartmentType.GameEngine = 'Unity'

Результат запроса:

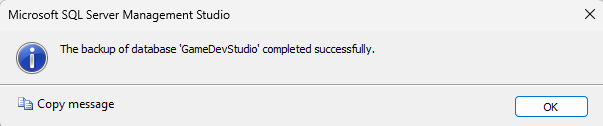


**7. Резервное копирование**

1) Окно создания резервной копии базы данных



2) Завершение создания резервной копии базы данных



**8. Инструкция для пользователя и для администратора**

1) Установка SQL Server

Ознакомьтесь с поддерживаемыми функциями для различных выпусков и версий SQL Server, чтобы определить, какие оптимальные варианты соответствуют вашим бизнес-потребностям. Проверьте требования к установке оборудования и программного обеспечения для SQL Server. Установите SQL Server с помощью графического интерфейса мастера установки, запущенного из файла setup.exe на установочном носителе.

2) Восстановление базы данных

В обозревателе объектов подключитесь к экземпляру компонента SQL Server Database Engine и разверните его. Нажмите правой кнопкой мыши узел базы данных и выберите команду «восстановить базу данных». Чтобы указать источник и расположение восстанавливаемых резервных наборов данных, используйте страницу Общие , раздел Источник . Выберите параметр «База данных». Измените имя базы данных по необходимости. Нажмите «ОК»

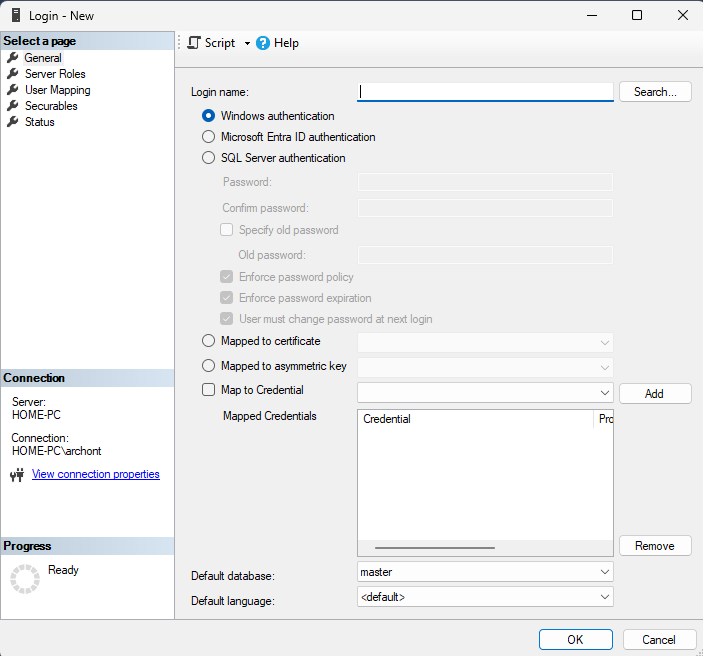
3) Установка ролей, привилегий и ограничений

Для того, чтобы пользоваться способом авторизации через логин и пароль, необходимо проделать следующие шаги:

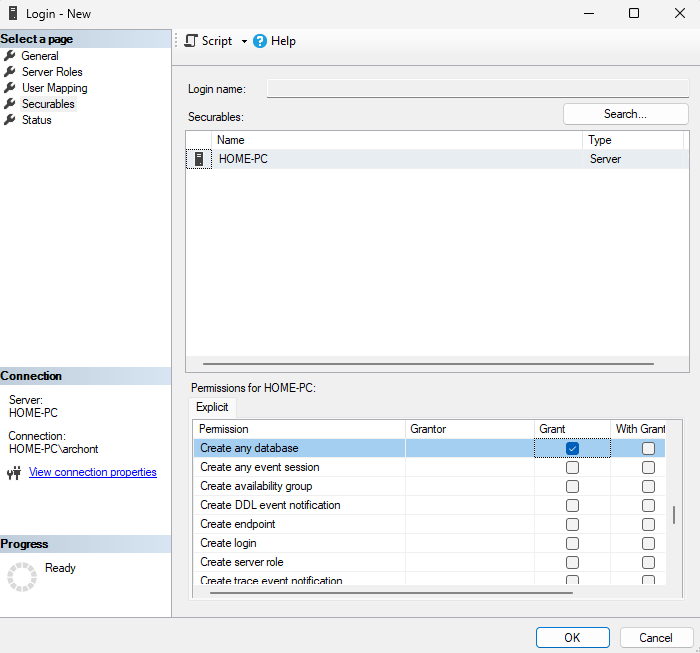
* Подключиться к локальному серверу
* Нажать правой кнопкой мыши на сервер и нажать на пункт «Properties»
* Во вкладке «Безопасность» на первых строках в пункте «Server authentification» выбрать пункт «SQL Server and Windows Authentication mode»
* Отключиться от сервера и перезапустить его в приложении SQL Server 2022 Configuration Manager

Все готово.

Далее необходимо снова подключиться к серверу. Чтобы создать роль, необходимо перейти на вкладку «Security». Внутри нее будет вкладка «Login», необходимо нажать на нее правой кнопкой мыши и нажать кнопку «New Login», откроется следующее окно:



Здесь необходимо в поле «Login name» указать имя пользователя, способ авторизации выбрать SQL Server Authentication, далее указать пароль, после этого перейти на вкладку «Server Roles» и выбрать необходимые роли для пользователя. При необходимости поставить дополнительные ограничения, нужно зайти на вкладку «Securables», где можно выбрать с помощью кнопки «Search» объекты, на которые необходимо задать ограничения, а в меню ниже — указать ограничения. Например, настройки на следующем фото позволят новому пользователю создавать новые базы данных на сервере «HOME-PC»:



По окончании нажать кнопку «ОК». Пользователь создан.