Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование и разработка баз данных интернет-приложений»

Отчёт по лабораторной работе №2

Реализация базы данных.

Выполнил: Корнелюк В. В.

ФИТ 3 курс 4 группа

Преподаватель: Нистюк О. Ф

Минск 2025

**Приложение для отслеживания исполнения**

Приложение предназначено для организаций и команд, которым необходимо эффективно отслеживать выполнение задач, поручений и проектов.

Определим физическую схему БД для СУБД Oracle – рисунок 1.

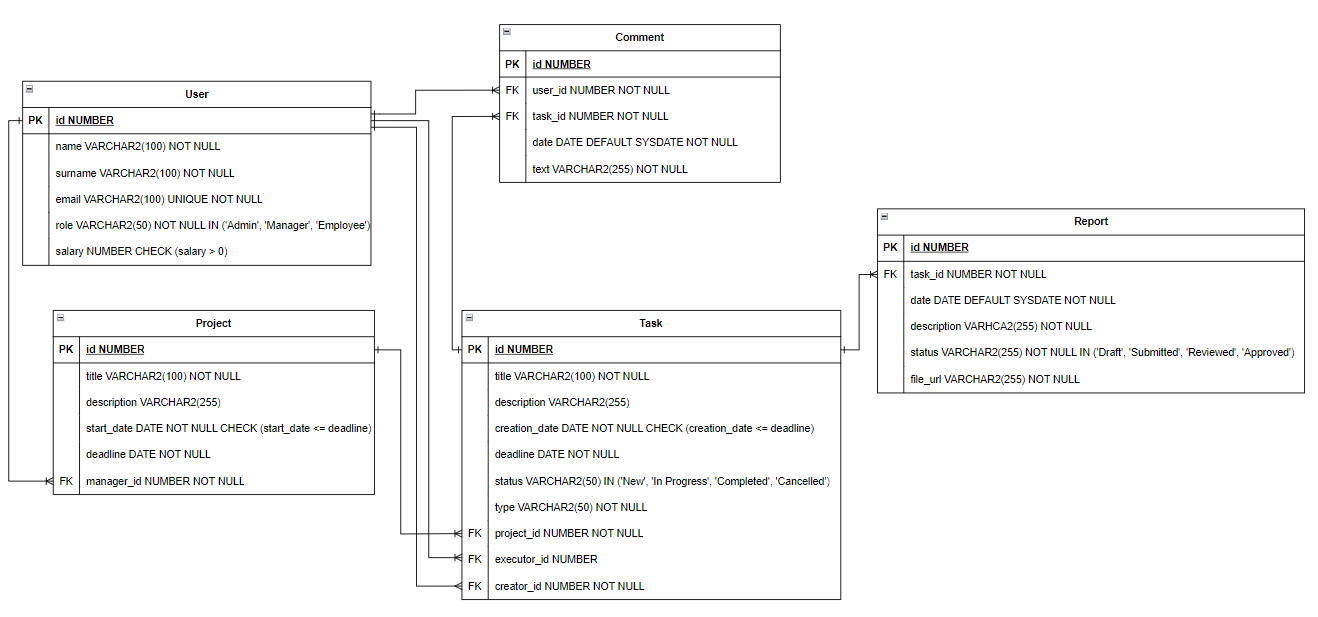


Рисунок 1 – Физическая схема БД для СУБД Oracle

Определим физическую схему БД для СУБД MS Server– рисунок 2.

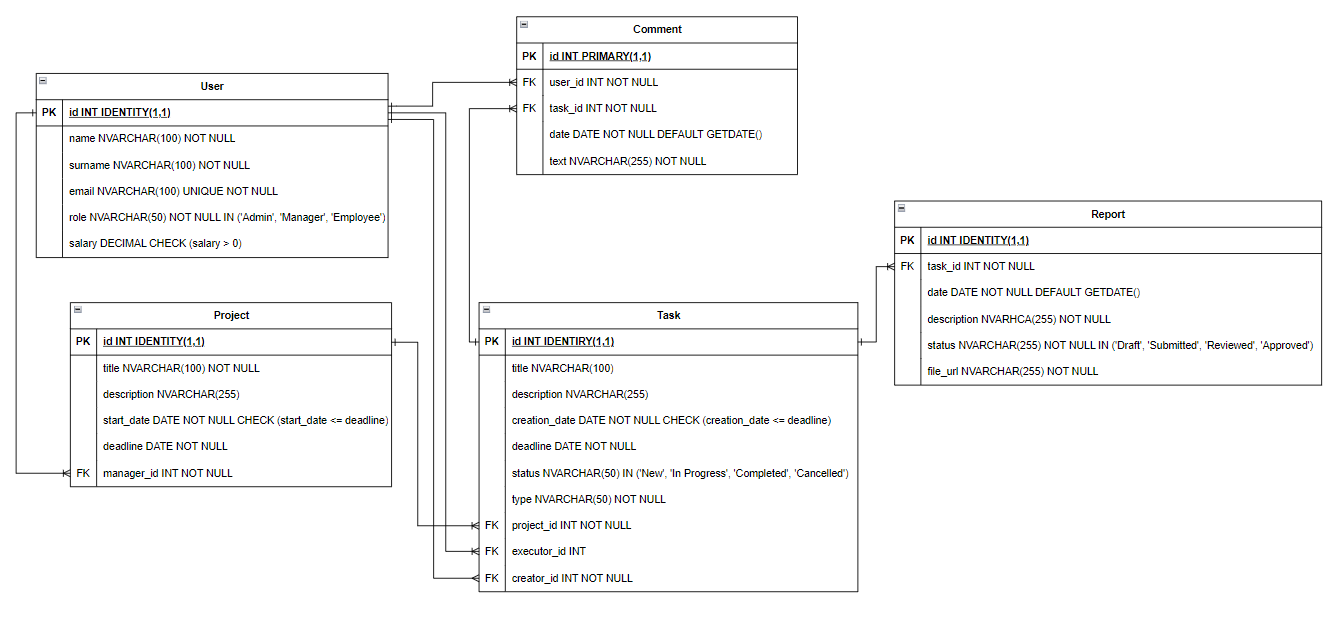


Рисунок 1 – Физическая схема БД для MS Server

**Таблицы SQL Server**

Таблица USER предназначена для хранения информации о пользователях. Таблица представлена в таблице 1.

Таблица 1 – таблица USERS

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | INT | PRIMARY KEY | Идентификатор пользователя |
| Name | NVARCHAR(100) | NOT NULL | Имя пользователя |
| Surname | NVARCHAR(100) | NOT NULL | Фамилия пользователя |
| Email | NVARCHAR(100) | UNIQUE NOT NULL | Электронная почта |
| Role | NVARCHAR(50) | NOT NULL | Роль пользователя |
| Salary | DECIMAL(18,2) | CHECK(Salary >= 0) | Зарплата |

Таблица PROJECT предназначена для хранения информации о проектах. Таблица представлена в таблице 2.

Таблица 2 – таблица PROJECT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | INT | PRIMARY KEY | Идентификатор проекта |
| Title | NVARCHAR(100) | NOT NULL | Название проекта |
| Description | NVARCHAR(255) |  | Описание проекта |
| Start\_date | DATE | NOT NULL | Дата начала проекта |
| Deadline | DATE | NOT NULL | Дедлайн |
| Manager\_id | INT | FOREIGN KEY | Идентификатор менеджера проекта |

Таблица TASK предназначена для хранения информации о задачах. Таблица представлена в таблице 3.

Таблица 3 – таблица TASK

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | INT | PRIMARY KEY | Идентификатор задачи |
| Title | NVARCHAR(100) | NOT NULL | Название задачи |
| Description | NVARCHAR(255) |  | Описание задачи |
| Creation\_date | DATE | NOT NULL | Дата создания задачи |
| Deadline | DATE | NOT NULL | Дедлайн |
| Type | NVARCHAR(50) | NOT NULL | Тип задачи |
| Project\_id | INT | FOREIGN KEY | Идентификатор проекта |
| Creator\_id | INT | FOREIGN KEY | Идентификатор создателя |
| Manager\_id | INT | FOREIGN KEY | Идентификатор менеджера проекта |

Таблица REPORT предназначена для хранения информации об очетах. Таблица представлена в таблице 4.

Таблица 4 – таблица REPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | INT | PRIMARY KEY | Идентификатор отчета |
| Date | DATE | NOT NULL | Дата создания |
| Description | NVARCHAR(255) | NOT NULL | Описание отчета |
| Status | NVARCHAR(50) | NOT NULL | Статус готовности отчета |
| File\_url | NVARCHAR(255) | NOT NULL | Расположение файла |

Таблица COMMENT предназначена для хранения комментариев. Таблица представлена в таблице 5.

Таблица 5 – таблица COMMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | INT | PRIMARY KEY | Идентификатор отчета |
| Task\_id | INT | FOREIGN KEY | Идентификатор задачи |
| User\_id | INT | FOREIGN KEY | Идентификатор пользователя |
| Date | DATE | NOT NULL | Дата создания |
| Text | NVARCHAR(255) | NOT NULL | Текст комментария |

**Таблицы Oracle**

Таблица USER предназначена для хранения информации о пользователях. Таблица представлена в таблице 6.

Таблица 6 – таблица USERS

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | NUMBER | PRIMARY KEY | Идентификатор пользователя |
| Name | VARCHAR2(100) | NOT NULL | Имя пользователя |
| Surname | VARCHAR2 (100) | NOT NULL | Фамилия пользователя |
| Email | VARCHAR2 (100) | UNIQUE NOT NULL | Электронная почта |
| Role | VARCHAR2 (50) | NOT NULL | Роль пользователя |
| Salary | NUMBER | CHECK(Salary >= 0) | Зарплата |

Таблица PROJECT предназначена для хранения информации о проектах. Таблица представлена в таблице 7.

Таблица 7 – таблица PROJECT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | NUMBER | PRIMARY KEY | Идентификатор проекта |
| Title | VARCHAR2 (100) | NOT NULL | Название проекта |
| Description | VARCHAR2 (255) |  | Описание проекта |
| Start\_date | DATE | NOT NULL | Дата начала проекта |
| Deadline | DATE | NOT NULL | Дедлайн |
| Manager\_id | NUMBER | FOREIGN KEY | Идентификатор менеджера проекта |

Таблица TASK предназначена для хранения информации о задачах. Таблица представлена в таблице 8.

Таблица 8 – таблица TASK

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | NUMBER | PRIMARY KEY | Идентификатор задачи |
| Title | VARCHAR2 (100) | NOT NULL | Название задачи |
| Description | VARCHAR2 (255) |  | Описание задачи |
| Creation\_date | DATE | NOT NULL | Дата создания задачи |
| Deadline | DATE | NOT NULL | Дедлайн |
| Type | VARCHAR2 (50) | NOT NULL | Тип задачи |
| Project\_id | NUMBER | FOREIGN KEY | Идентификатор проекта |
| Creator\_id | NUMBER | FOREIGN KEY | Идентификатор создателя |
| Manager\_id | NUMBER | FOREIGN KEY | Идентификатор менеджера проекта |

Таблица REPORT предназначена для хранения информации об очетах. Таблица представлена в таблице 9.

Таблица 9 – таблица REPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | NUMBER | PRIMARY KEY | Идентификатор отчета |
| Date | DATE | NOT NULL | Дата создания |
| Description | VARCHAR2 (255) | NOT NULL | Описание отчета |
| Status | VARCHAR2 (50) | NOT NULL | Статус готовности отчета |
| File\_url | VARCHAR2 (255) | NOT NULL | Расположение файла |

Таблица COMMENT предназначена для хранения комментариев. Таблица представлена в таблице 10.

Таблица 10 – таблица COMMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Тип данных | Ограничение | Назначение |
| Id | NUMBER | PRIMARY KEY | Идентификатор отчета |
| Task\_id | NUMBER | FOREIGN KEY | Идентификатор задачи |
| User\_id | NUMBER | FOREIGN KEY | Идентификатор пользователя |
| Date | DATE | NOT NULL | Дата создания |
| Text | VARCHAR2 (255) | NOT NULL | Текст комментария |

Определим процедуру insert\_user, которая представлена на листинге 1. Процедура принимает параметры name, surname, email, role, slary и предназначена для добавления нового пользователя.

**CREATE** **OR** **REPLACE** **PROCEDURE** insert\_user(

p\_name **VARCHAR2**,

p\_surname **VARCHAR2**,

p\_email **VARCHAR2**,

p\_role **VARCHAR2**,

p\_salary **NUMBER**

) **AS**

**BEGIN**

**INSERT** **INTO** **"USER"** (id, name, surname, email, **role**, salary)

**VALUES** (user\_seq.NEXTVAL, p\_name, p\_surname, p\_email, p\_role, p\_salary);

**COMMIT**;

**END**;

Листинг 1 – процедура insert\_user.

Определим процедуру insert\_project, которая представлена на листинге 2. Процедура принимает параметры title, description, start\_date, deadline, manager\_id и предназначена для добавления нового проекта.

**CREATE** **OR** **REPLACE** **PROCEDURE** insert\_project(

p\_title **VARCHAR2**,

p\_description **VARCHAR2**,

p\_start\_date **DATE**,

p\_deadline **DATE**,

p\_manager\_id **NUMBER**

) **AS**

**BEGIN**

**INSERT** **INTO** Project (id, title, description, start\_date, deadline, manager\_id)

**VALUES** (project\_seq.NEXTVAL, p\_title, p\_description, p\_start\_date, p\_deadline, p\_manager\_id);

**COMMIT**;

**END**;

Листинг 2 – процедура insert\_project.

Определим процедуру insert\_task, которая представлена на листинге 3. Процедура принимает параметры title, description, creation\_date, deadline, status, type, project\_id, creator\_id, executor\_id и предназначена для добавления новой задачи.

**CREATE** **OR** **REPLACE** **PROCEDURE** insert\_task(

p\_title **VARCHAR2**,

p\_description **VARCHAR2**,

p\_creation\_date **DATE**,

p\_deadline **DATE**,

p\_status **VARCHAR2**,

p\_type **VARCHAR2**,

p\_project\_id **NUMBER**,

p\_creator\_id **NUMBER**,

p\_executor\_id **NUMBER**

) **AS**

**BEGIN**

**INSERT** **INTO** Task (id, title, description, creation\_date, deadline, status, **type**, project\_id, creator\_id, executor\_id)

**VALUES** (task\_seq.NEXTVAL, p\_title, p\_description, p\_creation\_date, p\_deadline, p\_status, p\_type, p\_project\_id, p\_creator\_id, p\_executor\_id);

**COMMIT**;

**END**;

Листинг 3 – процедура insert\_task.

Определим процедуру insert\_report, которая представлена на листинге 4. Процедура принимает параметры task\_id, date, description, status, file\_url и предназначена для добавления нового отчета.

**CREATE** **OR** **REPLACE** **PROCEDURE** insert\_report(

p\_task\_id **NUMBER**,

p\_date **DATE**,

p\_description **VARCHAR2**,

p\_status **VARCHAR2**,

p\_file\_url **VARCHAR2**

) **AS**

**BEGIN**

**INSERT** **INTO** Report (id, task\_id, **"date"**, description, status, file\_url)

**VALUES** (report\_seq.NEXTVAL, p\_task\_id, p\_date, p\_description, p\_status, p\_file\_url);

**COMMIT**;

**END**;

Листинг 4 – процедура insert\_report.

Определим процедуру insert\_comment, которая представлена на листинге 5. Процедура принимает параметры task\_id, user\_id, date, text, и предназначена для добавления нового комментария.

**CREATE** **OR** **REPLACE** **PROCEDURE** insert\_comment(

p\_task\_id **NUMBER**,

p\_user\_id **NUMBER**,

p\_date **DATE**,

p\_text **VARCHAR2**

) **AS**

**BEGIN**

**INSERT** **INTO** **"COMMENT"** (id, task\_id, user\_id, **"date"**, text)

**VALUES** (comment\_seq.NEXTVAL, p\_task\_id, p\_user\_id, p\_date, p\_text);

**COMMIT**;

**END**;

Листинг 5 – процедура insert\_comment.

Определим процедуру get\_users, которая представлена на листинге 6. Процедура предназначена для добавления вывода пользователей.

**CREATE** **OR** **REPLACE** **PROCEDURE** get\_users **AS**

**BEGIN**

**FOR** rec **IN** (**SELECT** id, name, surname, email, **role**, salary **FROM** **"USER"**) **LOOP**

DBMS\_OUTPUT.PUT\_LINE(rec.id || ' | ' || rec.name || ' | ' || rec.surname || ' | ' || rec.email || ' | ' || rec.**role** || ' | ' || rec.salary);

**END** **LOOP**;

**END**;

Листинг 6 – процедура get\_users.

Определим процедуру get\_projects, которая представлена на листинге 7. Процедура предназначена для добавления вывода проектов.

**CREATE** **OR** **REPLACE** **PROCEDURE** get\_projects **AS**

**BEGIN**

**FOR** rec **IN** (**SELECT** id, title, description, start\_date, deadline, manager\_id **FROM** Project) **LOOP**

DBMS\_OUTPUT.PUT\_LINE(rec.id || ' | ' || rec.title || ' | ' || rec.description || ' | ' || rec.start\_date || ' | ' || rec.deadline || ' | ' || rec.manager\_id);

**END** **LOOP**;

**END**;

Листинг 7 – процедура get\_projects.

Определим процедуру get\_tasks, которая представлена на листинге 8. Процедура предназначена для добавления вывода задач.

**CREATE** **OR** **REPLACE** **PROCEDURE** get\_tasks **AS** **BEGIN**

**FOR** rec **IN** (**SELECT** id, title, description, creation\_date, deadline, status, **type**, project\_id, creator\_id, executor\_id **FROM** Task) **LOOP**

DBMS\_OUTPUT.PUT\_LINE(rec.id || ' | ' || rec.title || ' | ' || rec.description || ' | ' || rec.creation\_date || ' | ' || rec.deadline || ' | ' || rec.status || ' | ' || rec.**type** || ' | ' || rec.project\_id || ' | ' || rec.creator\_id || ' | ' || rec.executor\_id);

**END** **LOOP**; **END**;

Листинг 8 – процедура get\_tasks.

Определим процедуру get\_reports, которая представлена на листинге 9. Процедура предназначена для добавления вывода отчетов.

**CREATE** **OR** **REPLACE** **PROCEDURE** get\_reports **AS**

**BEGIN**

**FOR** rec **IN** (**SELECT** id, task\_id, **"date"**, description, status, file\_url **FROM** Report) **LOOP**

DBMS\_OUTPUT.PUT\_LINE(rec.id || ' | ' || rec.task\_id || ' | ' || rec.**"date"** || ' | ' || rec.description || ' | ' || rec.status || ' | ' || rec.file\_url);

**END** **LOOP**;

**END**;

Листинг 9 – процедура get\_reports.

Определим процедуру get\_comments, которая представлена на листинге 10. Процедура предназначена для добавления вывода комментрариев.

**CREATE** **OR** **REPLACE** **PROCEDURE** get\_comments **AS**

**BEGIN**

**FOR** rec **IN** (**SELECT** id, task\_id, user\_id, **"date"**, text **FROM** **"COMMENT"**) **LOOP**

DBMS\_OUTPUT.PUT\_LINE(rec.id || ' | ' || rec.task\_id || ' | ' || rec.user\_id || ' | ' || rec.**"date"** || ' | ' || rec.text);

**END** **LOOP**;

**END**;

Листинг 10 – процедура get\_comments.

Далее определим аналогичные процедуры для СУБД MS Sql.

Определим процедуру insert\_user, которая представлена на листинге 11. Процедура принимает параметры name, surname, email, role, slary и предназначена для добавления нового пользователя.

**CREATE** **PROCEDURE** insert\_user

**@name** **NVARCHAR**(100),

**@surname** **NVARCHAR**(100),

**@email** **NVARCHAR**(100),

**@role** **NVARCHAR**(50),

**@salary** **DECIMAL**(18,2)

**AS**

**BEGIN**

**INSERT** **INTO** **[User]** (name, surname, email, **role**, salary)

**VALUES** (**@name**, **@surname**, **@email**, **@role**, **@salary**); **END**;

Листинг 11 – процедура insert\_user.

Определим процедуру insert\_project, которая представлена на листинге 12. Процедура принимает параметры title, description, start\_date, deadline, manager\_id и предназначена для добавления нового проекта.

**CREATE** **PROCEDURE** insert\_project

**@title** **NVARCHAR**(100),

**@description** **NVARCHAR**(255),

**@start\_date** **DATE**,

**@deadline** **DATE**,

**@manager\_id** **INT**

**AS**

**BEGIN**

**INSERT** **INTO** **[Project]** (title, description, start\_date, deadline, manager\_id)

**VALUES** (**@title**, **@description**, **@start\_date**, **@deadline**, **@manager\_id**); **END**;

Листинг 12 – процедура insert\_project.

Определим процедуру insert\_task, которая представлена на листинге 13. Процедура принимает параметры title, description, creation\_date, deadline, status, type, project\_id, creator\_id, executor\_id и предназначена для добавления новой задачи.

**CREATE** **PROCEDURE** insert\_task

**@title** **NVARCHAR**(100),

**@description** **NVARCHAR**(255),

**@creation\_date** **DATE**,

**@deadline** **DATE**,

**@status** **NVARCHAR**(50),

**@type** **NVARCHAR**(50),

**@project\_id** **INT**,

**@creator\_id** **INT**,

**@executor\_id** **INT** **NULL** **AS**

**BEGIN**

**INSERT** **INTO** **[Task]** (title, description, creation\_date, deadline, status, **type**, project\_id, creator\_id, executor\_id)

**VALUES** (**@title**, **@description**, **@creation\_date**, **@deadline**, **@status**, **@type**, **@project\_id**, **@creator\_id**, **@executor\_id**);**END**;

Листинг 13 – процедура insert\_task.

Определим процедуру insert\_report, которая представлена на листинге 14. Процедура принимает параметры task\_id, date, description, status, file\_url и предназначена для добавления нового отчета.

**CREATE** **PROCEDURE** insert\_report

**@task\_id** **INT**,

**@date** **DATE**,

**@description** **NVARCHAR**(255),

**@status** **NVARCHAR**(50),

**@file\_url** **NVARCHAR**(255) **NULL**

**AS**

**BEGIN**

**INSERT** **INTO** **[Report]** (task\_id, **date**, description, status, file\_url)

**VALUES** (**@task\_id**, **@date**, **@description**, **@status**, **@file\_url**);

**END**;

Листинг 14 – процедура insert\_report.

Определим процедуру insert\_comment, которая представлена на листинге 15. Процедура принимает параметры task\_id, user\_id, date, text, и предназначена для добавления нового комментария.

**CREATE** **PROCEDURE** insert\_comment

**@task\_id** **INT**,

**@user\_id** **INT**,

**@date** **DATE**,

**@text** **NVARCHAR**(255)

**AS**

**BEGIN**

**INSERT** **INTO** **[Comment]** (task\_id, user\_id, **date**, **text**)

**VALUES** (**@task\_id**, **@user\_id**, **@date**, **@text**);

**END**;

Листинг 15 – процедура insert\_comment.

Определим процедуру get\_users, которая представлена на листинге 16. Процедура предназначена для добавления вывода пользователей.

**CREATE** **OR** **ALTER** **PROCEDURE** get\_users

**AS**

**BEGIN**

**DECLARE** **@output** **NVARCHAR**(**MAX**);

**DECLARE** cur **CURSOR** **FOR**

**SELECT** id, name, surname, email, **role**, salary **FROM** **[USER]**;

**DECLARE** **@id** **INT**, **@name** **NVARCHAR**(100), **@surname** **NVARCHAR**(100), **@email** **NVARCHAR**(255), **@role** **NVARCHAR**(50), **@salary** **DECIMAL**(10,2);

**OPEN** cur;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@name**, **@surname**, **@email**, **@role**, **@salary**;

**WHILE** **@@FETCH\_STATUS** = 0

**BEGIN**

**SET** **@output** = **CAST**(**@id** **AS** **NVARCHAR**) + ' | ' + **@name** + ' | ' + **@surname** + ' | ' + **@email** + ' | ' + **@role** + ' | ' + **CAST**(**@salary** **AS** **NVARCHAR**);

**PRINT** **@output**;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@name**, **@surname**, **@email**, **@role**, **@salary**;

**END**;

**CLOSE** cur;

**DEALLOCATE** cur;

**END**;

Листинг 16 – процедура get\_users.

Определим процедуру get\_projects, которая представлена на листинге 17. Процедура предназначена для добавления вывода проектов.

**CREATE** **OR** **ALTER** **PROCEDURE** get\_projects

**AS**

**BEGIN**

**DECLARE** **@output** **NVARCHAR**(**MAX**);

**DECLARE** cur **CURSOR** **FOR**

**SELECT** id, title, description, start\_date, deadline, manager\_id **FROM** Project;

**DECLARE** **@id** **INT**, **@title** **NVARCHAR**(255), **@description** **NVARCHAR**(500), **@start\_date** **DATE**, **@deadline** **DATE**, **@manager\_id** **INT**;

**OPEN** cur;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@title**, **@description**, **@start\_date**, **@deadline**, **@manager\_id**;

**WHILE** **@@FETCH\_STATUS** = 0

**BEGIN**

**SET** **@output** = **CAST**(**@id** **AS** **NVARCHAR**) + ' | ' + **@title** + ' | ' + **@description** + ' | ' + **CAST**(**@start\_date** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@deadline** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@manager\_id** **AS** **NVARCHAR**);

**PRINT** **@output**;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@title**, **@description**, **@start\_date**, **@deadline**, **@manager\_id**;

**END**;

**CLOSE** cur;

**DEALLOCATE** cur;

**END**;

Листинг 17 – процедура get\_projects.

Определим процедуру get\_tasks, которая представлена на листинге 18. Процедура предназначена для добавления вывода задач.

**CREATE** **OR** **ALTER** **PROCEDURE** get\_tasks

**AS**

**BEGIN**

**DECLARE** **@output** **NVARCHAR**(**MAX**);

**DECLARE** cur **CURSOR** **FOR**

**SELECT** id, title, description, creation\_date, deadline, status, **type**, project\_id, creator\_id, executor\_id **FROM** Task;

**DECLARE** **@id** **INT**, **@title** **NVARCHAR**(255), **@description** **NVARCHAR**(500), **@creation\_date** **DATE**, **@deadline** **DATE**, **@status** **NVARCHAR**(50), **@type** **NVARCHAR**(50), **@project\_id** **INT**, **@creator\_id** **INT**, **@executor\_id** **INT**;

**OPEN** cur;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@title**, **@description**, **@creation\_date**, **@deadline**, **@status**, **@type**, **@project\_id**, **@creator\_id**, **@executor\_id**;

**WHILE** **@@FETCH\_STATUS** = 0

**BEGIN**

**SET** **@output** = **CAST**(**@id** **AS** **NVARCHAR**) + ' | ' + **@title** + ' | ' + **@description** + ' | ' + **CAST**(**@creation\_date** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@deadline** **AS** **NVARCHAR**) + ' | ' + **@status** + ' | ' + **@type** + ' | ' + **CAST**(**@project\_id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@creator\_id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@executor\_id** **AS** **NVARCHAR**);

**PRINT** **@output**;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@title**, **@description**, **@creation\_date**, **@deadline**, **@status**, **@type**, **@project\_id**, **@creator\_id**, **@executor\_id**;

**END**;

**CLOSE** cur;

**DEALLOCATE** cur;

**END**;

Листинг 18 – процедура get\_tasks.

Определим процедуру get\_reports, которая представлена на листинге 19. Процедура предназначена для добавления вывода отчетов.

**CREATE** **OR** **ALTER** **PROCEDURE** get\_reports

**AS**

**BEGIN**

**DECLARE** **@output** **NVARCHAR**(**MAX**);

**DECLARE** cur **CURSOR** **FOR**

**SELECT** id, task\_id, **[date]**, description, status, file\_url **FROM** Report;

**DECLARE** **@id** **INT**, **@task\_id** **INT**, **@date** **DATE**, **@description** **NVARCHAR**(500), **@status** **NVARCHAR**(50), **@file\_url** **NVARCHAR**(500);

**OPEN** cur;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@task\_id**, **@date**, **@description**, **@status**, **@file\_url**;

**WHILE** **@@FETCH\_STATUS** = 0

**BEGIN**

**SET** **@output** = **CAST**(**@id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@task\_id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@date** **AS** **NVARCHAR**) + ' | ' + **@description** + ' | ' + **@status** + ' | ' + **@file\_url**;

**PRINT** **@output**;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@task\_id**, **@date**, **@description**, **@status**, **@file\_url**;

**END**;

**CLOSE** cur;

**DEALLOCATE** cur;

**END**;

Листинг 19 – процедура get\_reports.

Определим процедуру get\_comments, которая представлена на листинге 20. Процедура предназначена для добавления вывода комментрариев.

**CREATE** **OR** **ALTER** **PROCEDURE** get\_comments

**AS**

**BEGIN**

**DECLARE** **@output** **NVARCHAR**(**MAX**);

**DECLARE** cur **CURSOR** **FOR**

**SELECT** id, task\_id, user\_id, **[date]**, **text** **FROM** **[COMMENT]**;

**DECLARE** **@id** **INT**, **@task\_id** **INT**, **@user\_id** **INT**, **@date** **DATE**, **@text** **NVARCHAR**(1000);

**OPEN** cur;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@task\_id**, **@user\_id**, **@date**, **@text**;

**WHILE** **@@FETCH\_STATUS** = 0

**BEGIN**

**SET** **@output** = **CAST**(**@id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@task\_id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@user\_id** **AS** **NVARCHAR**) + ' | ' + **CAST**(**@date** **AS** **NVARCHAR**) + ' | ' + **@text**;

**PRINT** **@output**;

**FETCH** **NEXT** **FROM** cur **INTO** **@id**, **@task\_id**, **@user\_id**, **@date**, **@text**;

**END**;

**CLOSE** cur;

**DEALLOCATE** cur;

**END**;

Листинг 20 – процедура get\_comments.

Далее определим представление EmployeeTasks, которое представлено на листинге 21. Процедура предназначено для работы с активными задачами исполнителя.

**CREATE** **OR** **REPLACE** **VIEW** EmployeeTasks **AS**

**SELECT**

u.id **AS** employee\_id,

u.name **AS** employee\_name,

u.surname **AS** employee\_surname,

u.email **AS** employee\_email,

u.**role** **AS** employee\_role,

u.salary **AS** employee\_salary,

t.id **AS** task\_id,

t.title **AS** task\_title,

t.status **AS** task\_status,

t.description **AS** task\_description,

t.creation\_date,

t.deadline,

t.status,

t.**type**,

t.project\_id

**FROM** **"USER"** u

**JOIN** Task t **ON** u.id = t.executor\_id

**WHERE** t.status **in** ('New', 'In Progress');

Листинг 21 – представление EmployeeTasks.

Определим аналогичное представление EmployeeTasks для СУБД MS Sql, которое представлено в листинге 22. Процедура предназначено для работы с активными задачами исполнителя.

**CREATE** **OR** **ALTER** **VIEW** EmployeeTasks **AS**

**SELECT**

u.id **AS** employee\_id,

u.name **AS** employee\_name,

u.surname **AS** employee\_surname,

u.email **AS** employee\_email,

u.**role** **AS** employee\_role,

u.salary **AS** employee\_salary,

t.id **AS** task\_id,

t.title **AS** task\_title,

t.status **AS** task\_status,

t.description **AS** task\_description,

t.creation\_date,

t.deadline,

t.status,

t.**type**,

t.project\_id

**FROM** **[USER]** u

**JOIN** Task t **ON** u.id = t.executor\_id

**WHERE** t.status **in** ('New', 'In Progress');

Листинг 22 – представление EmployeeTasks.

Далее определим последовательность user\_seq, которая представлена в листинге 23. Последовательность предназначена для генерации идентификаторов для таблицы пользователей.

**CREATE** **SEQUENCE** user\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

NOCACHE

NOCYCLE;

Листинг 23 – последовательность user\_seq.

Также определим триггер user\_before\_insert, который представлен н листинге 24. Триггер предназначен для добавления идентификатора для таблицы пользователей при добавлении новой записи.

**CREATE** **OR** **REPLACE** **TRIGGER** user\_before\_insert

**BEFORE** **INSERT** **ON** **"USER"**

**FOR** **EACH** **ROW**

**BEGIN**

**IF** **:NEW**.id **IS** **NULL** **THEN**

**SELECT** user\_seq.NEXTVAL **INTO** **:NEW**.id **FROM** dual;

**END** **IF**;

**END**;

Листинг 24 – тиггер user\_before\_insert.

Определим индекс idx­\_report\_date, который представлен н листинге 25. Индекс предназначен для ускорения работы с таблицей отчетов.

**CREATE** **INDEX** idx\_report\_date **ON** Report (**"date"**);

Листинг 25 – индекс idx\_report\_date.

Определим аналогичный индекс idx­\_report\_date для СУБД MS Sql, который представлен в листинге 26. Индекс предназначен для ускорения работы с таблицей отчетов.

**CREATE** **INDEX** idx\_report\_date **ON** Report (**"date"**);

Листинг 26 – индекс idx\_report\_date.

Определим функцию индекс get\_project\_completion\_rate, которая представлена в листинге 27. Функция предназначена для расчета процента выполненных задач на проекте.

**CREATE** **OR** **REPLACE** **FUNCTION** get\_project\_completion\_rate(p\_project\_id **NUMBER**)

**RETURN** **NUMBER**

**IS**

total\_tasks **NUMBER**;

completed\_tasks **NUMBER**;

completion\_rate **NUMBER**;

**BEGIN**

**SELECT** **COUNT**(\*) **INTO** total\_tasks

**FROM** Task

**WHERE** project\_id = p\_project\_id;

**SELECT** **COUNT**(\*) **INTO** completed\_tasks

**FROM** Task

**WHERE** project\_id = p\_project\_id **AND** status = 'Completed';

**IF** total\_tasks = 0 **THEN**

**RETURN** 0;

**ELSE**

completion\_rate := (completed\_tasks / total\_tasks) \* 100;

**RETURN** completion\_rate;

**END** **IF**;

**END**;

Листинг 27 – функция get\_project\_completion\_rate.

Определим аналогичную функцию индекс get\_project\_completion\_rate для СУБД MS Sql, которая представлена в листинге 28. Функция предназначена для расчета процента выполненных задач на проекте.

**CREATE** **FUNCTION** get\_project\_completion\_rate(**@project\_id** **INT**)

**RETURNS** **FLOAT**

**AS**

**BEGIN**

**DECLARE** **@total\_tasks** **INT**, **@completed\_tasks** **INT**;

**SELECT** **@total\_tasks** = **COUNT**(\*) **FROM** Task **WHERE** project\_id = **@project\_id**;

**SELECT** **@completed\_tasks** = **COUNT**(\*) **FROM** Task **WHERE** project\_id = **@project\_id** **AND** status = 'Completed';

**IF** **@total\_tasks** = 0

**RETURN** 0;

**RETURN** (**CAST**(**@completed\_tasks** **AS** **FLOAT**) / **@total\_tasks**) \* 100;

**END**;

Листинг 28 – функция get\_project\_completion\_rate.

**Вывод**: была разработана физическая схема базы данных, созданы базы данных для СУБД Oracle и SQL Server, также были созданы необходимые для базы данных объекты.