Методические указания
по лабораторной работе №2
на тему: «Реализация БД в рамках СУБД»

по дисциплине: Информационная безопасность баз данных

Санкт-Петербург
2024

Цель работы

Получение навыков по работе с современными системами управления базами данных.

Задание

1. Выбрать систему управления базами данных (СУБД), которая будет использована в рамках лабораторной работы. Кратко обосновать свой выбор.
2. Создать БД в выбранной в вами СУБД на основе итоговой разработанной схемы отношений из ЛР 1. Заполните созданную вами БД информацией, сгенерируйте как минимум 7-8 кортежей с данными для каждой из ваших основных таблиц. В отчете по лабораторной работе укажите следующий SQL-код (написанный вами или сгенерированный средствами администрирования СУБД):
   - код для создания всех таблиц;
   - код для внесения данных в созданные таблицы;
   - код хотя бы одной SQL-команды для модифицирования структуры таблицы;
3. Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.
4. Установить взаимосвязи между таблицами.
5. **Дополнительно. Тестовых запросов к вашей БД**
6. Создать представления, составленные в пункте 5 лабораторной 1.

Пример отчета по лабораторной работе №2.

1. В качестве инструментария для создания БД использована СУБД PostgreSQL, а также графический интерфейс pgAdmin4. Используемая ссылка для скачивания https://www.postgresql.org/download/. Данные инструмент был выбран, поскольку это является одним из требований заказчика.
2. Создадим БД, согласно разработанной в ЛР1 схеме отношений. Для этого используем графический интерфейс pgAdmin. Создадим базу данных, в которой будут находиться все отношения

```
CREATE DATABASE dbs24
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
```

Для начала создадим сущности в соответствии со схемой отношений на рисунке 1.

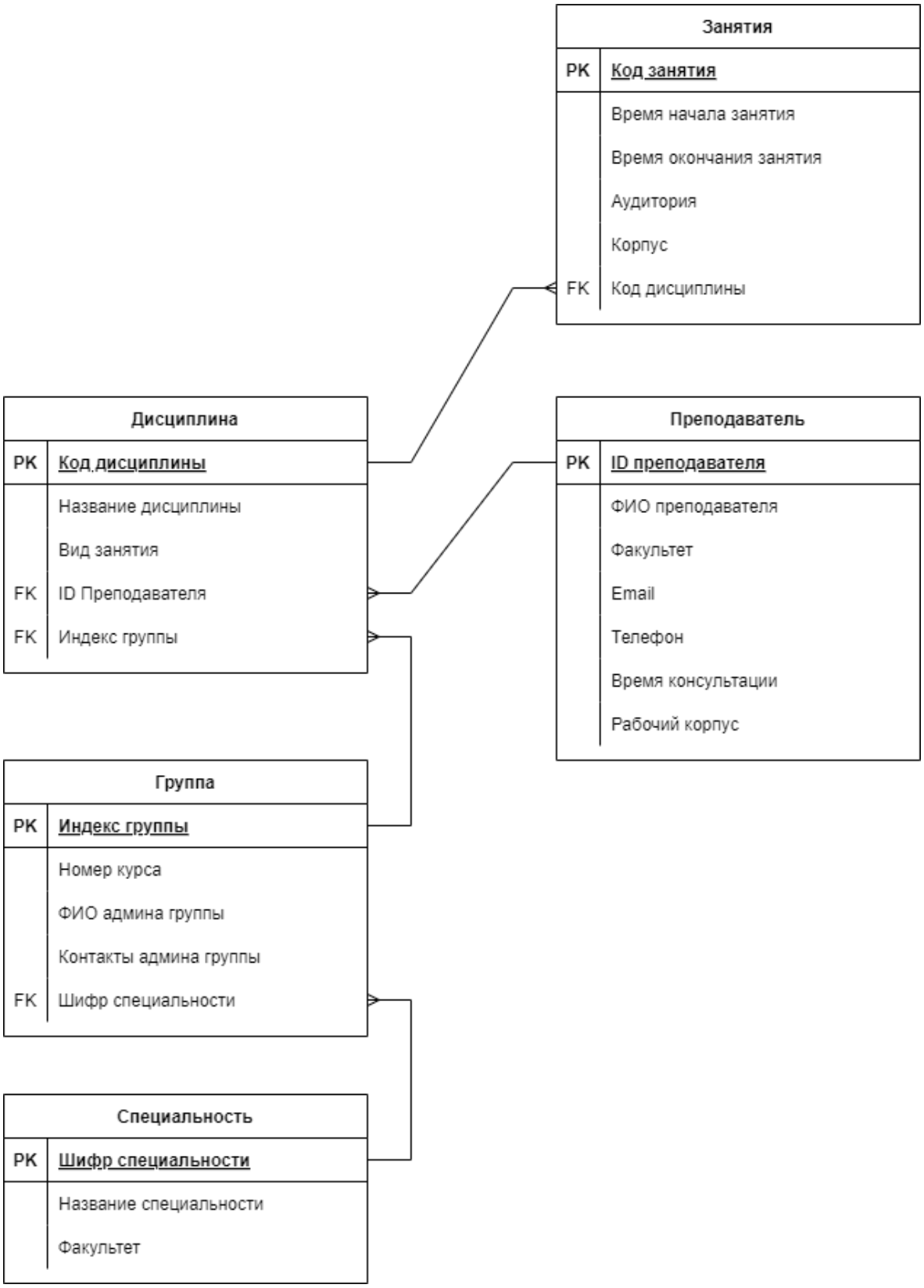

Рисунок 1 – Схема предварительных отношений БД университета

Создадим таблицы, соответствующие каждому из отношений схемы предварительных отношений:

CREATE TABLE IF NOT EXISTS public.classes

```
(
    classes_id integer NOT NULL,
    classes_start_time text,
    classes_end_time text,
    classroom text,
    building text,
    subject_id integer,
    CONSTRAINT classes_pkey PRIMARY KEY (classes_id)
)

CREATE TABLE IF NOT EXISTS public.subject
(
    subject_id integer NOT NULL,
    subject_name text,
    subject_type text,
    teacher_id integer NOT NULL,
    group_id integer NOT NULL,
    CONSTRAINT subject_pkey PRIMARY KEY (subject_id)
)

CREATE TABLE IF NOT EXISTS public.teachers
(
    teacher_id integer NOT NULL,
    teacher_full_name text,
    department text,
    email text,
    phone text,
    consultation_time text,
    working_bulding text,
    CONSTRAINT teachers_pkey PRIMARY KEY (teacher_id)
)

CREATE TABLE IF NOT EXISTS public.groups
(
    group_id integer NOT NULL,
    course_number integer,
    admin_full_name text,
    admin_contacts text,
    specialty_code integer NOT NULL,
    CONSTRAINT groups_pkey PRIMARY KEY (group_id)
)

CREATE TABLE IF NOT EXISTS public.specialty
(
    specialty_code integer NOT NULL,
    specialty_name text,
```

```
    department text,
    CONSTRAINT specialty_pkey PRIMARY KEY (specialty_code)
)
```

Добавим дополнительные ограничения на отношения, соответствующие внешним ключам и проиндексируем их, чтобы запросы при объединении таблиц выполнялись быстрее.

```
ALTER TABLE IF EXISTS public.classes
    ADD CONSTRAINT fk_class_subj FOREIGN KEY (subject_id)
    REFERENCES public.subject (subject_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID;
CREATE INDEX IF NOT EXISTS fki_k
    ON public.classes(subject_id);

ALTER TABLE IF EXISTS public.subject
    ADD CONSTRAINT fk_teacher_subj FOREIGN KEY (teacher_id)
    REFERENCES public.teachers (teacher_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID;
CREATE INDEX IF NOT EXISTS "fki_A"
    ON public.subject(teacher_id);

ALTER TABLE IF EXISTS public.subject
    ADD CONSTRAINT fk_group_subj FOREIGN KEY (group_id)
    REFERENCES public.groups (group_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID;
CREATE INDEX IF NOT EXISTS fki_d
    ON public.subject(group_id);

ALTER TABLE IF EXISTS public.groups
    ADD CONSTRAINT fk_spec_group FOREIGN KEY (specialty_code)
    REFERENCES public.specialty (specialty_code) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID;
CREATE INDEX IF NOT EXISTS "fki_ы"
    ON public.groups(specialty_code);
```

Заполним созданную схему данных информацией. Начнем заполнение с главных сущностей «Преподаватели» и «Специальность».

INSERT INTO public.teachers (teacher_id, teacher_full_name, department, email, phone, consultation_time, working_bulding)
VALUES
('192423'::integer, 'Kozlov Genadii Andreevich'::text, 'PBKS'::text, 'kga@gmail.com'::text, '(901)3235511'::text, '12-15, every monday'::text, 'Kronverskii, 14, classroom 404'::text),
('3277122'::integer, 'Vasykov Vladimir Vasilievich'::text, 'TSG'::text, 'vvv@mail.ru'::text, '(922)34783823'::text, '8-14, friday'::text, 'Pesochnaya, 14, class 300'::text),
('827241'::integer, ' Dementiev Sergey Danilovich '::text, ' KTU '::text, ' dsd@gmail.com '::text, '(911)8463244'::text, '18-20, every monday '::text, ' Birzhevaya, 16, classroom 412'::text);

Данные для сущности «Специальность»:

INSERT INTO public.specialty (specialty_code, specialty_name, department)
VALUES
('103114'::integer, 'Information Security'::text, 'KTU'::text),
('106288'::integer, 'IoT Security'::text, 'TSG'::text);

Вставка данных в отношение «Группа»:

INSERT INTO public.groups (group_id, course_number, admin_full_name, admin_contacts, specialty_code)
VALUES
('3351'::integer, '3'::integer, 'Vasiliev Andrei'::text, 'phone 921-323-23-44'::text, '103114'::integer),
('2311'::integer, '2'::integer, 'Krivov Dmitrii'::text, 'mail kdmitr@gmail.com'::text, '103114'::integer),
('1131'::integer, '1'::integer, 'Ivanov Ivan'::text, 'ivan@mail.ru'::text, '106288'::integer);

Генерация данных для отношения «Дисциплина»:

INSERT INTO public.subject (subject_id, subject_name, subject_type, teacher_id, group_id)
VALUES
('1'::integer, 'Database Security'::text, 'lection'::text, '192423'::integer, '1131'::integer),
('2'::integer, 'Database Security'::text, 'labs'::text, '827241'::integer, '1131'::integer),
('3'::integer, 'Cryptography'::text, 'lection'::text, '827241'::integer, '2311'::integer),
('4'::integer, 'Network Security'::text, 'labs'::text, '3277122'::integer, '2311'::integer),
('5'::integer, 'Application Security'::text, 'lection'::text, '3277122'::integer, '3351'::integer),
('6'::integer, 'Coding Theory'::text, 'lection'::text, '3277122'::integer, '3351'::integer)

Добавим несколько кортежей в отношение «Занятия»:

INSERT INTO public.classes (classes_id, classes_start_time, classes_end_time, classroom, building, subject_id)
VALUES
('1'::integer, '8.20'::text, '10.00'::text, '303'::text, 'Pesochnaya, 14'::text, '1'::integer),
('2'::integer, '11.00'::text, '11.30'::text, '304'::text, 'Pesochnaya, 14'::text, '1'::integer),
('3'::integer, '11.40'::text, '13.10'::text, '101'::text, 'Kronverskii, 16'::text, '2'::integer),
('4'::integer, '15.00'::text, '16.30'::text, '511'::text, 'Birzhevaya, 6'::text, '2'::integer),
('5'::integer, '18.00'::text, '19.30'::text, '513'::text, 'Kronverskii, 16'::text, '3'::integer),

('6'::integer, '8.20'::text, '11.40'::text, '404'::text, 'Pesochnaya, 14'::text, '4'::integer),
('7'::integer, '10.00'::text, '13.30'::text, '112'::text, 'Birzhevaya, 6'::text, '5'::integer),
('8'::integer, '11.00'::text, '12.00'::text, '112'::text, 'Pesochnaya, 14'::text, '5'::integer),
('9'::integer, '10.00'::text, '12.00'::text, '513'::text, 'Kronverskii, 16'::text, '6'::integer)

Создадим представления, составленные в пункте 5 лабораторной 1, для удобного просмотра информации пользователями.

Представление 1. «Компактное расписание», состоит из следующих атрибутов.

- Название дисциплины (отношение «Дисциплина»)
- Вид занятий (отношение «Дисциплина»)
- Время начала занятия (отношение «Занятия»)
- Время окончания занятия (отношение «Занятия»)
- Корпус (отношение «Занятия»)
- ФИО преподавателя (отношение «Преподаватель»)

Исходя из текущих обозначений атрибутов в БД, sql-код для создания представления может выглядеть следующим образом:

```
CREATE VIEW public.compact_shedule
 AS
select
        subject.subject_name,
        subject.subject_type,
        classes.classes_start_time,
        classes.classes_end_time,
        classes.building,
        teachers.teacher_full_name
from public.subject
JOIN public.classes ON subject.subject_id=classes.subject_id
JOIN public.teachers ON subject.teacher_id=teachers.teacher_id;
```

Представление 2. «Консультации и контакты преподавателей»
Список атрибутов, отображаемых в рамках представления «Консультации и контакты преподавателей»:

- Название дисциплины (отношение «Дисциплина»)
- ФИО преподавателя (отношение «Преподаватель»)
- Факультет (отношение «Преподаватель»)
- Email (отношение «Преподаватель»)
- Телефон (отношение «Преподаватель»)
- Время консультации (отношение «Преподаватель»)
- Рабочий корпус (отношение «Преподаватель»)

```
CREATE VIEW public.consultations
 AS
select
```

```
    subject.subject_name,
    teachers.teacher_full_name,
    teachers.department,
    teachers.email,
    teachers.phone,
    teachers.consultation_time,
    teachers.working_bulding
from public.subject
JOIN public.teachers ON subject.teacher_id=teachers.teacher_id;
```

Представление 3. «Специальности и контактные данные групп»
Список атрибутов, отображаемых в рамках представления «Специальности и контактные данные групп»:

- Название специальности (отношение «Специальность»)
- Индекс группы (отношение «Группа»)
- Номер курса (отношение «Группа»)
- ФИО админа группы (отношение «Группа»)
- Контакты админа группы (отношение «Группа»)

```
CREATE VIEW public.speciality_and_contacts
 AS
select
    specialty.specialty_name,
    groups.group_id,
    groups.course_number,
    groups.admin_full_name,
    groups.admin_contacts
from public.specialty
JOIN public.groups ON specialty.specialty_code=groups.specialty_code;
```

Представление 4. «Преподаватели, задействованные в рамках специальности»
Список атрибутов, отображаемых в рамках представления «Преподаватели, задействованные в рамках специальности»»:

- Название специальности (отношение «Специальность»)
- Факультет (отношение «Специальность»)
- Индекс группы (отношение «Группа»)
- Название дисциплины (отношение «Дисциплина»)
- ФИО преподавателя (отношение «Преподаватель»)

Код создания представления:

```
CREATE VIEW public.speciality_and_teachers
AS
select
    specialty.specialty_name,
    specialty.department,
    groups.group_id,
```

```
        subject.subject_name,
        teachers.teacher_full_name
from public.subject
JOIN public.teachers ON subject.teacher_id=teachers.teacher_id
JOIN public.groups ON subject.group_id=groups.group_id
JOIN public.specialty ON specialty.specialty_code=groups.specialty_code;
```