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;

1. Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.
2. Установить взаимосвязи между таблицами.
3. **Дополнительно. Тестовых запросов к вашей БД**
4. 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.

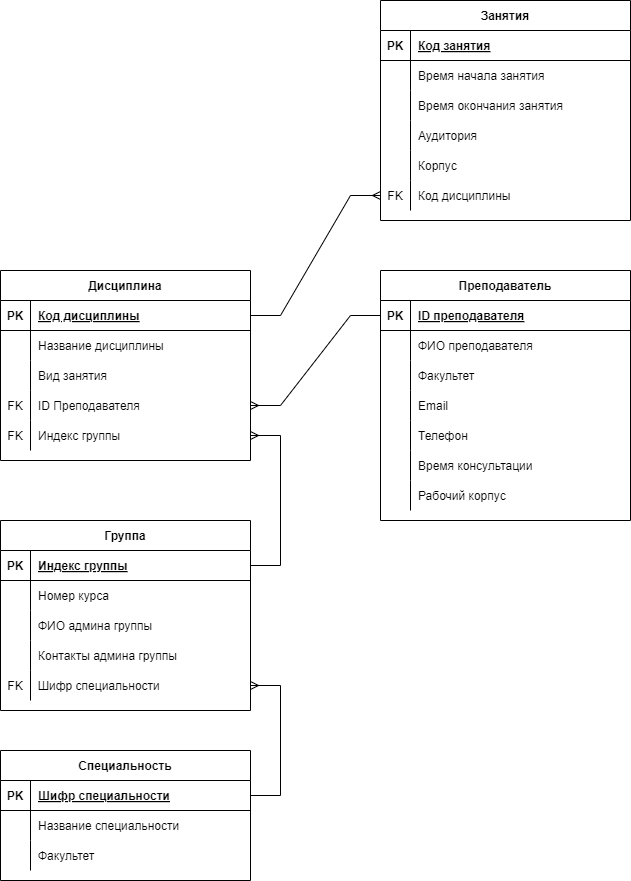


Рисунок 1 – Схема предварительных отношений БД университета

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\_А"*

*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;*