**Лабораторная работа №2**

**Тема: «Основы работы с СУБД PostgreSQL»**

Цель: овладеть основными практическими навыками типового взаимодействия с использованием интегрированных утилит СУБД с БД на примере PostgreSQL.

**2.1. Основные понятия реляционных баз данных**

**Упражнения**

1. Напишите запрос к таблице SUBJECT, выводящий для каждой ее строки идентификатор (номер) предмета обучения, его наименование, семестр, в котором он читается, и количество отводимых на него часов.

SELECT "SUBJ\_ID", "SUBJ\_NAME", "SEMESTR", "HOUR"

FROM "SUBJECT"

2. Напишите запрос, позволяющий вывести -все строки таблицы EXAM\_MARKS, в которых предмет обучения имеет номер (SUBJ\_ID), равный 12.

SELECT \*

FROM "EXAM\_MARKS"

WHERE "SUBJ\_ID" = 12

3. Напишите запрос, выбирающий все данные из таблицы STUDENT, расположив столбцы таблицы в следующем порядке: KURS, SURNAME, NАМЕ, STIPEND.

SELECT "KURS", "SURNAME", "NAME", "STIPEND"

FROM "STUDENT"

4. Напишите запрос SELECT, который для каждого предмета обучения (SUBJECT) выполняет вывод его наименования (SUBJ\_NAMЕ) и следом за ним количества часов (HOUR) в 4-м семестре (SEMESTR).

SELECT "SUBJ\_NAME", "HOUR", "SEMESTR"

FROM "SUBJECT"

WHERE "SEMESTR" = 4

5. Напишите запрос, позволяющий получить из таблицы EXAM\_MARKS значения столбца МАRК (экзаменационная оценка) для всех студентов, исключив из списка повторение одинаковых строк.

SELECT DISTINCT "STUDENT\_ID", "MARK"

FROM "EXAM\_MARKS"

6. Напишите запрос, который выполняет вывод списка фамилий студентов, обучающихся на третьем и более старших курсах.

SELECT "SURNAME", "KURS"

FROM "STUDENT"

WHERE "KURS" >=3

7. Напишите запрос, выбирающий данные фамилию, имя и номер курса для студентов, получающих стипендию больше 140.

SELECT "SURNAME", "NAME", "KURS"

FROM "STUDENT"

WHERE "STIPEND" > 140

8. Напишите запрос, выполняющий выборку нз таблицы SUBJECT названий всех предметов обучения, на которые отводится более 30 часов.

SELECT "SUBJ\_NAME"

FROM "SUBJECT"

WHERE "HOUR" > 30

9. Напишите запрос, который выполняет вывод.списка университетов, рейтинг которых превышает 300 баллов.

SELECT "UNIV\_NAME", "RATING"

FROM "UNIVERSITY"

WHERE "RATING" > 300

10. Напишите запрос к таблице STUDENT для вывода списка всех студентов со стипендией не меньше 100, живущих в Воронеже, с указанием фамилии (SURNAME), имени (NAME) и номера курса (KURS)

SELECT "SURNAME", "NAME", "KURS"

FROM "STUDENT"

WHERE "STIPEND" >=100 AND "CITY" = 'Воронеж'

11. Напишите запрос для получения списка студентов старше 25 лет, обучающихся на 1-м курсе.

SELECT \*

FROM "STUDENT"

WHERE "KURS" = 1 AND "BIRTHDAY" <= '1990-12-12'

12. Напишите запрос для получения списка предметов, для которых в 1-м семестре отведено более 100 часов.

SELECT "SUBJ\_NAME"

FROM "SUBJECT"

WHERE "SEMESTR" = 1 AND "HOUR" >= 100

13. Напишите запрос для получения списка преподавателей, живущих в Воронеже.

SELECT \*

FROM "LECTURER"

WHERE "CITY" = 'Воронеж'

14. Напишите запрос для получения списка университетов, расположенных в Москве и имеющих рейтинг меньший, чем у ВГУ. Константу в ограничении на рейтинг можно определить по этой же таблице.

SELECT \*

FROM "LECTURER"

WHERE "CITY" = 'Воронеж'

15. Напишите запрос для получения списка студентов, проживающих в Воронеже и не получающих стипендию.

SELECT \*

FROM "UNIVERSITY"

WHERE "CITY" = 'Москва' AND "RATING" > 296

16. Напишите запрос для получения списка студентов моложе 20 лет.

SELECT \*

FROM "STUDENT"

WHERE (extract(year from current\_date) - extract (year from("BIRTHDAY"))) > 30

17. Напишите запрос для получения списка студентов без определенного места жительства.

SELECT \*

FROM "STUDENT"

WHERE "CITY" IS NULL

**2.2 Операторы IN, BETWEEN, LIKE, IS NULL**

**Упражнения**

1. Напишите запрос, выполняющий вывод находящихся в таблице EXAM\_MARKS номеров предметов обучения, экзамены по которым сдавались между 10 и 20 января 2005 г.

SELECT "SUBJ\_ID"

FROM "EXAM\_MARKS"

WHERE "EXAM\_DATE" BETWEEN '2005-01-10' AND '2005-01-20'

2. Напишите запрос, выбирающий данные обо всех предметах обучения, экзамены по которым сданы студентами, имеющими идентификаторы 12 и 32.

SELECT \*

FROM "EXAM\_MARKS"

WHERE "STUDENT\_ID" IN (12, 32)

3. Напишите запрос, который выполняет вывод названий. предметов обучения, начинающихся на букву 'И'-;

SELECT "SUBJ\_NAME"

FROM "SUBJECT"

WHERE "SUBJ\_NAME" LIKE 'И%'

4. Напишите запрос, выбирающий сведения о студентах, у которых имена начинаются на букву 'И' или 'С'.

SELECT \*

FROM "STUDENT"

WHERE "NAME" LIKE 'И%' OR "NAME" LIKE 'С%'

5. Напишите запрос для выбора из таблицы EXAM\_MARKS записей, для которых отсутствуют значения оценок (поле MARK).

SELECT \*

FROM "EXAM\_MARKS"

WHERE "MARK" IS NULL

6. Напишите запрос, выполняющий вывод из таблицы EXAM\_MARKS записей, для которых в поле MARK проставлены значения оценок.

SELECT \*

FROM "EXAM\_MARKS"

WHERE "MARK" IS NOT NULL

7. Напишите запрос для получения списка преподавателей, проживающих в городах, в названиях которых присутствует дефис.

SELECT \*

FROM "LECTURER"

WHERE "CITY" LIKE '%-%'

8. Напишите запрос для получения списка предметов, названия которых оканчиваются на 'ия'.

SELECT \*

FROM "SUBJECT"

WHERE "SUBJ\_NAME" LIKE '%ия'

9. Напишите запрос для получения списка студентов, фамилии которых начинаются на 'Ков' или на 'Куз'.

SELECT \*

FROM "STUDENT"

WHERE "SURNAME" LIKE 'Ков%' OR "SURNAME" LIKE 'Куз%'

10. Напишите запрос для получения списка предметов обучения, названия которых состоят из более одного слова.

SELECT \*

FROM "SUBJECT"

WHERE "SUBJ\_NAME" LIKE '% %';

11. Напишите запрос для получения списка учебных заведений, аббревиатура которых состоит как минимум из 4 букв.

SELECT \*

FROM "UNIVERSITY"

WHERE "UNIV\_NAME" LIKE '\_\_\_\_%'

12. Напишите запрос для получения списка студентов, фамилии которых состоят из трех букв.

SELECT \*

FROM "STUDENT"

WHERE "SURNAME" LIKE '\_\_\_'

**2.3 Преобразование вывода и встроенные функции**

**Упражнения**

1. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала один столбец, содержащий последовательность разделенных символом ";" (точка с запятой) значений всех столбцов этой таблицы; при этом текстовые значения должны отображаться прописными символами (верхний регистр), т. е. быть представленными в следующем виде:

10; КУЗНЕЦОВ; БОРИС; 0; БРЯНСК; 8.12.1987; 10.

SELECT UPPER("STUDENT\_ID"||';'||"SURNAME"||';'||"NAME"||';'||"KURS"||';'||"CITY"||';'||"BIRTHDAY"||';'||"UNIV\_ID")

FROM "STUDENT"

2. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Б. КУЗНЕЦОВ; место жительства — БРЯНСК; родился — 8.12.87.

SELECT SUBSTR("NAME", 1,1)||'.'||UPPER("SURNAME")||';'||' место жительства'||' -- '||UPPER("CITY")||';'||' родился'||' -- '||

TO\_CHAR("BIRTHDAY",'DD.MM.YY')

FROM "STUDENT"

3. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: б. кузнецов; место жительства — брянск; родился: 8-дек-1987.

SELECT LOWER(SUBSTR("NAME", 1,1))||'.'||LOWER("SURNAME")||';'||' место жительства'||' -- '||LOWER("CITY")||';'||' родился'||': '||

TO\_CHAR("BIRTHDAY",'DD.Mon.YYYY')

FROM "STUDENT"

4. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Борис Кузнецов родился в 1987 году.

SELECT "NAME"||' '||"SURNAME"||' '|| 'родился в' ||' '|| TO\_CHAR("BIRTHDAY",'YYYY') ||' '||'году'

FROM "STUDENT"

5. Составьте запрос, выводящий фамилии, имена студентов и величину получаемых ими стипендий, при этом значения стипендий должны быть увеличены в 100 раз.

SELECT "SURNAME", "NAME", "STIPEND"\*100

FROM "STUDENT"

6. То же, что и в упр. 4, но только для студентов 1, 2 и 4 курсов и таким образом, чтобы фамилии и имена были выведены прописными буквами.

SELECT UPPER("NAME"||' '||"SURNAME")||' '|| 'родился в' ||' '|| TO\_CHAR("BIRTHDAY",'YYYY') ||' '||'году'

FROM "STUDENT"

WHERE "KURS" IN (1,2,4)

7 Составьте запрос для таблицы UNIVERSITY таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Код-10; ВГУ-r. ВОРОНЕЖ; Рейтинг=296.

SELECT 'Код'||'-'||"UNIV\_ID"||'; '||"UNIV\_NAME"||'-г. '||UPPER("CITY")||'; '||'Рейтинг'||'='||"RATING"

FROM "UNIVERSITY"

8. То же; что и в упр. 7, но значения рейтинга требуется округлить до первого знака (например, значение -382 округляется-до 400).

SELECT 'Код'||'-'||"UNIV\_ID"||'; '||"UNIV\_NAME"||'-г. '||UPPER("CITY")||'; '||'Рейтинг'||'='||ROUND("RATING",-2)

FROM "UNIVERSITY"

**2.4 Агрегирование и групповые функции**

**Упражнения**

1. Напишите запрос для подсчета количества студентов, сдавших (поле MARK непустое) экзамен по предмету обучения с идентификатором 12.

SELECT COUNT(\*)

FROM "EXAM\_MARKS"

WHERE "SUBJ\_ID" = 12 AND "MARK" IS NOT NULL;

2. Напишите запрос, который позволяет подсчитать в таблице EXAM\_MARKS количество различных предметов обучения (использовать оператор DISTINCT.

SELECT COUNT(DISTINCT "SUBJ\_ID")

FROM "EXAM\_MARKS";

3. Напишите запрос, который для каждого студента выполняет выборку его идентификатора и минимальной из полученных им оценок.

SELECT "STUDENT\_ID", MIN("MARK")

FROM "EXAM\_MARKS"

GROUP BY "STUDENT\_ID";

4. Напишите запрос, который для каждого студента выполняет выборку его идентификатора и максимальной из полученных им оценок.

SELECT "STUDENT\_ID", MAX("MARK")

FROM "EXAM\_MARKS"

GROUP BY "STUDENT\_ID";

6. Напишите запрос, который для каждого конкретного дня сдачи экзамена выводит данные о количестве студентов, сдававших экзамен в этот день.

SELECT "EXAM\_DATE", COUNT("STUDENT\_ID")

FROM "EXAM\_MARKS"

GROUP BY "EXAM\_DATE";

7. Напишите запрос, выдающий средний балл для каждого студента.

SELECT AVG("MARK")

FROM "EXAM\_MARKS"

GROUP BY "STUDENT\_ID"

8. Напишите запрос, выдающий средний балл для каждого экзамена.

SELECT AVG("MARK")

FROM "EXAM\_MARKS"

GROUP BY "SUBJ\_ID"

9. Напишите запрос, определяющий количество сдававших студентов для каждого экзамена.

SELECT COUNT(DISTINCT"STUDENT\_ID")

FROM "EXAM\_MARKS"

GROUP BY "SUBJ\_ID"

10. Напишите запрос для определения количества предметов, изучаемых на каждом курсе.

SELECT COUNT(DISTINCT "SUBJ\_ID")

FROM "EXAM\_MARKS", "STUDENT"

GROUP BY "KURS"

11. Для каждого университета напишите запрос, выводящий суммарную стипендию обучающихся в нем студентов, с последующей сортировкой списка по этому значению.

SELECT "UNIV\_ID", SUM("STIPEND")

FROM "STUDENT"

GROUP BY "UNIV\_ID"

12. Для каждого семестра напишите запрос, выводящий общее количество часов, отводимое на изучение соответствующих предметов.

SELECT "SEMESTR", SUM("HOUR")

FROM "SUBJECT"

GROUP BY "SEMESTR"

13. Для каждого студента напишите запрос, выводящий среднее значение оценок, полученных им по каждому предмету.

SELECT AVG("MARK")

FROM "EXAM\_MARKS"

GROUP BY "SUBJ\_ID", "STUDENT\_ID"

14. Напишите запрос, выводящий количество студентов, проживающих в каждом городе. Список отсортировать в порядке убывания количества студентов.

SELECT COUNT(DISTINCT "STUDENT\_ID")

FROM "STUDENT"

ORDER BY COUNT(DISTINCT "STUDENT\_ID") DESC

15. Для каждого университета напишите запрос, выводящий количество обучающихся в нем студентов, с последующей сортировкой списка по этому количеству.

SELECT COUNT("STUDENT\_ID"), "UNIV\_ID"

FROM "STUDENT"

GROUP BY "UNIV\_ID"

16. Для каждого университета напишите запрос, выводящий количество работающих в нем преподавателей, с последующей сортировкой списка по этому количеству.

SELECT COUNT("LECTURER\_ID"), "UNIV\_ID"

FROM "LECTURER"

GROUP BY "UNIV\_ID"

ORDER BY COUNT("LECTURER\_ID") DESC

17. Для каждого университета напишите запрос, выводящий сумму стипендии, выплачиваемой студентам каждого курса.

SELECT SUM("STIPEND"), "UNIV\_ID", "KURS"

FROM "STUDENT"

GROUP BY "UNIV\_ID", "KURS"

ORDER BY "UNIV\_ID", "KURS"

18. Для каждого города напишите запрос, выводящий максимальный рейтинг университетов, в нем расположенных, с последующей сортировкой списка по значениям рейтингов.

SELECT MAX("RATING"), "CITY"

FROM "UNIVERSITY"

GROUP BY "CITY"

19. Для каждого дня сдачи экзаменов напишите запрос, выводящий среднее значение всех экзаменационных оценок.

SELECT AVG("MARK"), "EXAM\_DATE"

FROM "EXAM\_MARKS"

GROUP BY "EXAM\_DATE"

20. Для каждого дня сдачи экзаменов напишите запрос, выводящий максимальные оценки, полученные по каждому предмету.

SELECT MAX("MARK"), "EXAM\_DATE"

FROM "EXAM\_MARKS"

GROUP BY "EXAM\_DATE"

21. Для каждого дня сдачи экзаменов напишите запрос, выводящий общее количество экзаменов, сдававшихся каждым студентом.

SELECT COUNT("SUBJ\_ID"), "EXAM\_DATE"

FROM "EXAM\_MARKS"

GROUP BY "EXAM\_DATE"

22. Для каждого преподавателя напишите запрос, выводящий количество преподаваемых им предметов.

SELECT COUNT("SUBJ\_ID"), "LECTURER\_ID"

FROM "SUBJ\_LECT"

GROUP BY "SUBJ\_ID", "LECTURER\_ID"

23. Для каждого предмета напишите запрос, выводящий количество преподавателей, ведущих по нему занятия

SELECT COUNT("LECTURER\_ID"), "SUBJ\_ID"

FROM "SUBJ\_LECT"

GROUP BY "SUBJ\_ID"

24. Напишите запрос, выполняющий вывод количества студентов, имеющих только отличные оценки.

SELECT COUNT("STUDENT\_ID")

FROM "EXAM\_MARKS"

WHERE "MARK" = 5

GROUP BY "MARK"

25. Напишите запрос, выполняющий вывод количества экзаменов; сданных (с положительной оценкой) студентом с идентификатором 32.

SELECT COUNT("SUBJ\_ID")

FROM "EXAM\_MARKS"

WHERE "MARK">=3 AND "STUDENT\_ID" = 32

**2.5.1 Вложенные подзапросы**

**Упражнения**

1. Напишите запрос, выводящий список студентов, получающих максимальную стипендию, отсортировав его в алфавитном порядке по фамилиям.

SELECT \*

FROM "STUDENT"

WHERE "STIPEND" =

(SELECT MAX("STIPEND")

FROM "STUDENT"

)

ORDER BY "SURNAME" ASC

2. Напишите запрос, выводящий список студентов, получающих стипендию, превышающую среднее значение стипендии.

SELECT \*

FROM "STUDENT"

WHERE "STIPEND">

(SELECT AVG("STIPEND")

FROM "STUDENT")

8. Напишите запрос, выводящий список, студентов, обучающихся в Воронеже, с последующей сортировкой по идентификаторам университетов и курсам.

SELECT \*

FROM "STUDENT"

WHERE "UNIV\_ID" IN

(SELECT "UNIV\_ID"

FROM "UNIVERSITY"

WHERE "CITY" = 'Воронеж')

ORDER BY "UNIV\_ID", "KURS"

9. Напишите запрос, выводящий список предметов, на изучение которых отведено максимальное количество часов.

SELECT \*

FROM "SUBJECT"

WHERE "HOUR" IN

(SELECT MAX("HOUR")

FROM "SUBJECT")

10. Напишите запрос, выполняющий вывод имен и фамилий студентов, место проживания которых не совпадает с городом, в котором находится их университет.

SELECT "NAME", "SURNAME"

FROM "STUDENT", "UNIVERSITY"

WHERE "STUDENT"."UNIV\_ID" = "UNIVERSITY"."UNIV\_ID" and "STUDENT"."CITY" <> "UNIVERSITY"."CITY"

11. Напишите запрос, выводящий список университетов, расположенных в Москве и имеющих рейтинг меньший, чем у ВГУ.

SELECT \*

FROM "UNIVERSITY"

WHERE "CITY" = 'Москва' AND "RATING" >

(SELECT "RATING"

FROM "UNIVERSITY"

WHERE "UNIV\_NAME" = 'ВГУ')

**2.5.2 Операции соединения таблиц посредством ссылочной целостности.**

**Упражнения**

1:. Напишите запрос для получения списка университетов с указанием количества студентов, обучающихся на каждом курсе.

SELECT "UNIV\_NAME", "KURS" ,COUNT("STUDENT\_ID")

FROM "UNIVERSITY" JOIN "STUDENT"

ON "UNIVERSITY"."UNIV\_ID" = "STUDENT"."UNIV\_ID"

GROUP BY "UNIV\_NAME", "KURS"

2. Напишите запрос для получения списка преподавателей с указанием их учебных предметов.

SELECT "SURNAME","SUBJECT"."SUBJ\_ID", "SUBJ\_NAME"

FROM "LECTURER" JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJECT"."SUBJ\_ID" = "SUBJ\_LECT"."SUBJ\_ID"

GROUP BY "SURNAME","SUBJECT"."SUBJ\_ID", "SUBJ\_NAME"

3. Напишите запрос для получения списка преподавателей с указанием нагрузки (суммарного количества часов) в каждом семестре.

SELECT "SURNAME", "HOUR", "SEMESTR"

FROM "LECTURER"

JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJ\_LECT"."SUBJ\_ID" = "SUBJECT"."SUBJ\_ID"

GROUP BY "SURNAME", "HOUR", "SEMESTR"

4. Напишите запрос для получения списка университетов вместе с названиями преподаваемых в них предметов.

SELECT "UNIV\_NAME", "SUBJ\_NAME"

FROM "UNIVERSITY"

JOIN "LECTURER"

ON "UNIVERSITY"."UNIV\_ID" = "LECTURER"."UNIV\_ID"

JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJ\_LECT"."SUBJ\_ID" = "SUBJECT"."SUBJ\_ID"

GROUP BY "UNIV\_NAME", "SUBJ\_NAME"

5. Напишите запрос для получения списка университетов с указанием суммарного количества аудиторных часов в каждом семестре.

SELECT "UNIV\_NAME", "HOUR", "SEMESTR"

FROM "UNIVERSITY"

JOIN "LECTURER"

ON "UNIVERSITY"."UNIV\_ID" = "LECTURER"."UNIV\_ID"

JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJ\_LECT"."SUBJ\_ID" = "SUBJECT"."SUBJ\_ID"

GROUP BY "UNIV\_NAME", "HOUR", "SEMESTR"

6. Напишите запрос для получения списка университетов с указанием суммарного количества часов, отводимых на изучение каждого предмета.

SELECT "UNIV\_NAME", "HOUR", "SUBJ\_NAME"

FROM "UNIVERSITY"

JOIN "LECTURER"

ON "UNIVERSITY"."UNIV\_ID" = "LECTURER"."UNIV\_ID"

JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJ\_LECT"."SUBJ\_ID" = "SUBJECT"."SUBJ\_ID"

GROUP BY "UNIV\_NAME", "HOUR", "SUBJ\_NAME"

7. Напишите запрос для получения списка преподавателей с указанием суммарного количества часов, отведенных для обучения каждому из предметов.

SELECT "SURNAME", "HOUR", "SUBJ\_NAME"

FROM "LECTURER"

JOIN "SUBJ\_LECT"

ON "LECTURER"."LECTURER\_ID" = "SUBJ\_LECT"."LECTURER\_ID"

JOIN "SUBJECT"

ON "SUBJ\_LECT"."SUBJ\_ID" = "SUBJECT"."SUBJ\_ID"

GROUP BY "SURNAME", "HOUR", "SUBJ\_NAME"

8. Напишите запрос для сортировки списка университетов по значениям максимальной стипендии, выплачиваемой студентам.

SELECT "UNIV\_NAME", MAX("STIPEND")

FROM "UNIVERSITY" JOIN "STUDENT"

ON "UNIVERSITY"."UNIV\_ID" = "STUDENT"."UNIV\_ID"

GROUP BY "UNIV\_NAME"

9. Напишите запрос для получения списка университетов вместе с фамилиями самых молодых студентов, обучающихся в них.

SELECT "UNIV\_NAME", MAX("BIRTHDAY")

FROM "UNIVERSITY" JOIN "STUDENT"

ON "UNIVERSITY"."UNIV\_ID" = "STUDENT"."UNIV\_ID"

GROUP BY "UNIV\_NAME"

10. Напишите запрос для получения списка университетов вместе с фамилиями студентов, получающих максимальную для каждого университета стипендию.

SELECT "UNIV\_NAME", "STIPEND", "SURNAME"

FROM "UNIVERSITY" JOIN "STUDENT"

ON "UNIVERSITY"."UNIV\_ID" = "STUDENT"."UNIV\_ID"

WHERE ("STUDENT"."UNIV\_ID", "STIPEND") IN

(SELECT "UNIV\_ID", MAX("STIPEND")

FROM "STUDENT"

GROUP BY "UNIV\_ID")

ORDER BY "UNIV\_NAME", "SURNAME"

**2.6 Создание, изменение добавление данных и таблиц**

**УПРАЖНЕНИЕ 1**

1. Напишите команду, которая вводит в таблицу SUBJECT строку для нового предмета обучения со следующими значениями полей: SEMESTER = 4; SUBJ\_NAME = 'Алгебра'; HOUR = 72; SUBJ\_ID = 201.

INSERT INTO "SUBJECT"("SEMESTR", "SUBJ\_NAME", "HOUR", "SUBJ\_ID")

VALUES ('4', 'Алгебра', '72', '201')

2. Введите запись для нового студента, которого зовут Орлов Николай, обучающегося на первом курсе ВГУ, живущего в Воронеже, сведения о дате рождения и размере стипендии неизвестны.

INSERT INTO "STUDENT"("SURNAME", "NAME", "KURS", "CITY", "UNIV\_ID")

VALUES ('Орлов', 'Николай', '1','Воронеж','10')

3. Напишите команду, удаляющую из таблицы EXAMMARKS записи обо всех оценках студента, идентификатор которого равен 100.

DELETE FROM "EXAM\_MARKS"

WHERE "STUDENT\_ID" = 100

4. Напишите команду, которая увеличивает на 5 значение рейтинга всех университетов, расположенных в Санкт-Петербурге.

UPDATE "UNIVERSITY"

SET "RATING" = "RATING"+5

WHERE "CITY" = 'Санкт-Петербург'

5. Измените в таблице значение города, в котором проживает студент Иванов, на 'Воронеж'.

UPDATE "STUDENT"

SET "CITY" = 'Воронеж'

WHERE "SURNAME" = 'Иванов'

**УПРАЖНЕНИЕ 2**

1. Пусть существует таблица с именем STUDENT1, определения столбцов которой полностью совпадают с определениями столбцов таблицы STUDENT. Вставьте в эту таблицу сведения о студентах, успешно сдавших экзамены более чем по пяти предметам обучения.

SELECT "STUDENT"."STUDENT\_ID", "STUDENT"."SURNAME", "STUDENT"."NAME", "STUDENT"."STIPEND", "STUDENT"."KURS", "STUDENT"."CITY", "STUDENT"."BIRTHDAY", "STUDENT"."UNIV\_ID"

FROM "STUDENT"

WHERE "STUDENT"."STUDENT\_ID" in (

SELECT "STUDENT\_ID"

FROM "EXAM\_MARKS"

WHERE "MARK" >=3

GROUP BY "STUDENT\_ID"

HAVING COUNT("MARK") > 1);

2. Напишите команду, удаляющую из таблицы SUBJECT1 сведения о предметах обучения, по которым студентами не получено ни одной оценки.

DELETE FROM "SUBJECT1"

WHERE "SUBJ\_ID" NOT IN

(SELECT DISTINCT("SUBJECT"."SUBJ\_ID")

FROM "EXAM\_MARKS", "SUBJECT"

WHERE "SUBJECT"."SUBJ\_ID" = "EXAM\_MARKS"."SUBJ\_ID" AND "MARK" IS NOT NULL

)

3. Напишите запрос, увеличивающий размер стипендии на 20% всем студентам, у которых общая сумма баллов превышает значение 50.

UPDATE "STUDENT"

SET "STIPEND" = ("STIPEND" + (0.2 \* "STIPEND"))

WHERE 50 <=

(SELECT SUM("MARK")

FROM "EXAM\_MARKS"

WHERE "EXAM\_MARKS"."STUDENT\_ID" = "STUDENT"."STUDENT\_ID" )

**УПРАЖНЕНИЕ 3**

1. Напишите команду CREATE TABLE для создания таблицы LECTURER1.

CREATE TABLE "LECTURER1"

("LECTURER\_ID" INTEGER,

"SURNAME" VARCHAR(60),

"NAME" VARCHAR (60),

"CITY" VARCHAR(60),

"UNIV\_ID" INTEGER);

2. Напишите команду CREATE TABLE для создания таблицы SUBJECT1

CREATE TABLE "SUBJ\_LECT2"

("SUBJ\_ID" INTEGER,

"SUBJ\_NAME" VARCHAR(60),

"KURS" INTEGER,

"SEMESTR" INTEGER);

3. Напишите команду CREATE TABLE для создания таблицы UNIVERSITY1.

CREATE TABLE "UNIVERSITY1"

("UNIV\_ID" INTEGER,

"UNIV\_NAME" VARCHAR(60),

"RATING" INTEGER,

"CITY" VARCHAR(60));

4. Напишите команду CREATE TABLE для создания таблицы EXAM\_MARKS1.

CREATE TABLE "EXAM\_MARKS1"

("EXAM\_ID" INTEGER,

"STUDENT\_ID" INTEGER,

"SUBJ\_ID" INTEGER,

"RATING" INTEGER,

"EXAM\_DATE" DATE);

5. Напишите команду CREATE TABLE для создания таблицы SUBJ\_LECT1.

CREATE TABLE "SUBJ\_LECT1"

("LECTURER\_ID" INTEGER,

"SUBJ\_ID" INTEGER);

**УПРАЖНЕНИЕ 4**

1. Создайте таблицу EXAM\_MARKS так, чтобы не допускался ввод в таблицу двух записей об оценках одного студента по конкретным экзамену и предмету обучения.

CREATE TABLE "EXAM\_MARKS\_NEW1"

("EXAM\_ID" INTEGER,

"STUDENT\_ID" INTEGER,

"SUBJ\_ID" INTEGER,

"MARK" INTEGER,

"EXAM\_DATE" DATE,

CONSTRAINT "EXAM\_OGR" PRIMARY KEY("STUDENT\_ID", "EXAM\_ID", "SUBJ\_ID"))

2. Создайте таблицу предметов обучения SUBJECT2 так, чтобы количество отводимых на предмет часов по умолчанию было равно 36, не допускались записи с отсутствующим количеством часов, поле SUBJ\_ID являлось первичным ключом таблицы, а значения семестров (поле SEMESTR) лежали в диапазоне от 1 до 12.

CREATE TABLE "SUBJECT2"

("SUBJ\_ID" INTEGER PRIMARY KEY,

"SUBJ\_NAME" VARCHAR,

"HOUR" INTEGER NOT NULL DEFAULT '36',

"SEMESTR" INTEGER CHECK("SEMESTR" IN (1,12)))