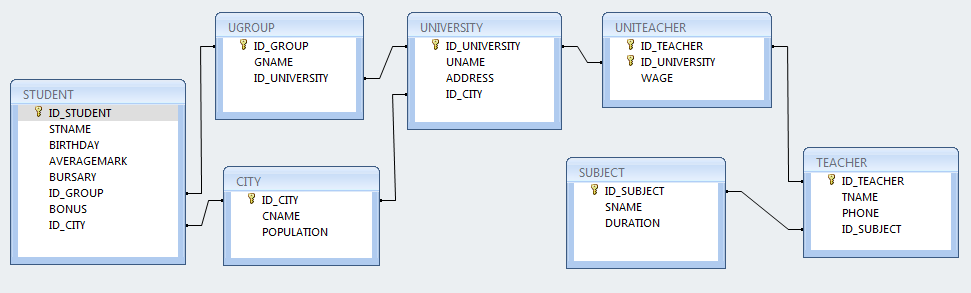
**Учебная база данных и тестовые задания**

ER-диаграмма учебной базы:



**Результат выполнения тестового задания – набор SQL-запросов для Oracle. Просто поместите ваши ответы в этот документ! Задания, присланные в другом формате, проверяться не будут!**

**Отлаживать запросы можно с помощью любой из утилит.**

Тестовые задания и ожидаемые результаты:

1. Выбрать студентов из таблицы STUDENT, и место их проживания из таблицы CITY . Показать только тех студентов, которые учатся в университете по улице Бровки (Brovki) и отсортировать их по имени (Команда SELECT с условием, Несколько простых JOINs, LEFT OUTER JOIN, сортировка). 1 sql-команда.

SELECT STUDENT.STNAME, CITY.CNAME

FROM STUDENT

JOIN UGROUP on UGROUP.ID\_GROUP = STUDENT.ID\_GROUP

JOIN UNIVERSITY on UNIVERSITY.ID\_UNIVERSITY = UGROUP.ID\_UNIVERSITY

LEFT JOIN CITY on CITY.ID\_CITY = STUDENT.ID\_CITY

WHERE ADDRESS LIKE '%BROVKI%'

ORDER BY STNAME ASC



1. Показать все города и университеты, которые находятся в этих городах, отсортировать результат по номеру университета (LEFT JOIN). 1 sql-команда.

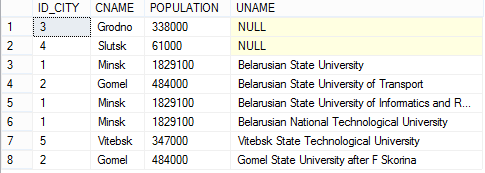
SELECT CITY.ID\_CITY, CITY.CNAME, CITY.POPULATION, UNIVERSITY.UNAME

FROM STUDENT

JOIN CITY ON CITY.ID\_CITY = STUDENT.ID\_CITY

LEFT JOIN UNIVERSITY ON UNIVERSITY.ID\_CITY = CITY.ID\_CITY

ORDER BY ID\_UNIVERSITY ASC



1. Показать все предметы и преподавателей, которые их преподают, и среднюю зарплату каждого преподавателя по университетам (LEFT JOINS, групповая функция AVG, CONVERT). 1 sql-команда.

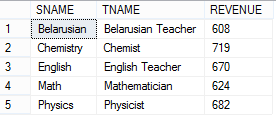
SELECT TEACHER.TNAME, SUBJECT.SNAME, CONVERT (int, AVG(WAGE)) AS 'REVENUE'

FROM TEACHER

JOIN UNITEACHER ON UNITEACHER.ID\_TEACHER = TEACHER.ID\_TEACHER

JOIN SUBJECT ON SUBJECT.ID\_SUBJECT = TEACHER.ID\_SUBJECT

GROUP BY TEACHER.TNAME, SUBJECT.SNAME



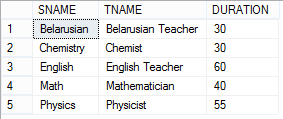
1. Показать всех преподавателей, предметы, которые они преподают, и их длительность (RIGHT JOIN). 1 sql-команда.

SELECT SUBJECT.SNAME, TEACHER.TNAME, SUBJECT.DURATION

FROM TEACHER

RIGHT JOIN SUBJECT ON SUBJECT.ID\_SUBJECT = TEACHER.ID\_SUBJECT

ORDER BY SNAME



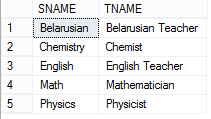
1. Показать всех преподавателей и все предметы (FULL JOIN). 1 sql-команда.

SELECT SUBJECT.SNAME, TEACHER.TNAME

FROM TEACHER

FULL JOIN SUBJECT ON SUBJECT.ID\_SUBJECT = TEACHER.ID\_SUBJECT

ORDER BY SNAME ASC



1. Подсчитать среднее число жителей, проживающих в городах из таблицы City (групповая функция AVG, CONVERT). 1 sql-команда.

SELECT CONCAT ('The average amount of inhabitans is ', CONVERT (int, AVG (POPULATION)), 'humans') AS 'The average amount' FROM CITY



1. Выбрать самого молодого студента из таблицы STUDENT (групповая функция MAX, подзапрос в условии). 1 sql-команда.

SELECT STNAME, BIRTHDAY

FROM STUDENT

WHERE BIRTHDAY = (SELECT MAX(Birthday) FROM STUDENT)



1. Показать все города и количество университетов в каждом городе (псевдостолбец-подзапрос, псевдонимы столбцов, групповая функция COUNT). 1 sql-команда.

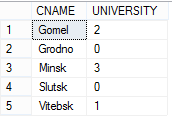
SELECT CNAME,

(SELECT COUNT (UNIVERSITY.ID\_CITY) from UNIVERSITY

WHERE (UNIVERSITY.ID\_CITY = CITY.ID\_CITY)) AS 'UNIVERSITY'

FROM CITY

ORDER BY CNAME



1. Выбрать города, в которых нет студентов, которые учатся в университетах из таблицы UNIVERSITY (конструкция EXISTS, подзапрос на таблицу STUDENT и простые JOINs, CONVERT, COALESCE). 1 sql-команда.

SELECT \* FROM CITY

WHERE NOT EXISTS (SELECT CONVERT (int, COALESCE (ID\_STUDENT,0))

FROM STUDENT

JOIN UGROUP ON UGROUP.ID\_GROUP = STUDENT.ID\_GROUP

JOIN UNIVERSITY ON UNIVERSITY.ID\_UNIVERSITY = UGROUP.ID\_UNIVERSITY

WHERE STUDENT.ID\_CITY=CITY.ID\_CITY)



1. Показать средний доход (стипендию (BURSARY) + премию (BONUS)) студентов по группам из города Минска (Minsk) из виртуальной таблицы, которая является результатом объединения всех столбцов из таблиц STUDENT и CITY. Отсортировать группы по убыванию дохода. (Групповая функция AVG, подзапрос c простым JOIN в секции FROM) .1 sql-команда.

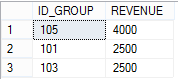
SELECT STUDENT.ID\_GROUP, CONVERT (INT, AVG (BURSARY + COALESCE(BONUS,0))) AS 'REVENUE'

FROM STUDENT, CITY

WHERE CITY.CNAME = 'Minsk' AND STUDENT.ID\_CITY = CITY.ID\_CITY

GROUP BY STUDENT.ID\_GROUP, BURSARY, BONUS

ORDER BY REVENUE DESC



1. Подсчитать среднюю стипендию по университетам и город, в котором он находиться. Показать только те университеты, у которых суммарная сумма на выплаты студентам стипендии и премий больше 10000 (Несколько простых JOINs, групповые функции AVG, SUM, фильтр групп HAVING, преобразование NULL в конкретное значение). 1 sql-команда.

SELECT CITY.CNAME, UNIVERSITY.UNAME, CONVERT (int, AVG (COALESCE (STUDENT.BURSARY,0)))   
AS 'AVERAGE BURSARY'

FROM CITY

JOIN UNIVERSITY ON UNIVERSITY.ID\_CITY = CITY.ID\_CITY

JOIN UGROUP ON UGROUP.ID\_UNIVERSITY = UNIVERSITY.ID\_UNIVERSITY

JOIN STUDENT ON STUDENT.ID\_GROUP = UGROUP.ID\_GROUP

GROUP BY CITY.CNAME, UNIVERSITY.UNAME

HAVING SUM (COALESCE (STUDENT.BURSARY,0)) > 10000



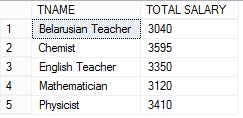
1. Подсчитать суммарный доход преподавателей по университетам (групповая функция SUM, LEFT JOIN). 1 sql-команда.

SELECT TNAME, SUM (WAGE) AS 'TOTAL SALARY'

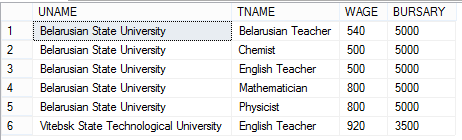
FROM TEACHER

LEFT JOIN UNITEACHER ON UNITEACHER.ID\_TEACHER = TEACHER.ID\_TEACHER

GROUP BY TNAME



1. Показать университеты, в которых преподают самые дорогостоящие преподаватели (исходя из их заработной платы) ИЛИ самые богатые студенты (исходя из их стипендии) (Несколько простых JOINS, подзапросы в условии, групповая функция MAX, DISTINCT) . 1 sql-команда.



SELECT DISTINCT UNAME, TNAME, WAGE, BURSARY

FROM UNIVERSITY

JOIN UNITEACHER ON UNITEACHER.ID\_UNIVERSITY = UNIVERSITY.ID\_UNIVERSITY

JOIN TEACHER ON TEACHER.ID\_TEACHER = UNITEACHER.ID\_TEACHER

JOIN UGROUP ON UGROUP.ID\_UNIVERSITY = UNIVERSITY.ID\_UNIVERSITY

JOIN STUDENT ON STUDENT.ID\_GROUP = UGROUP.ID\_GROUP

WHERE WAGE = (SELECT MAX (WAGE) FROM UNITEACHER)

OR BURSARY = (SELECT MAX (BURSARY) FROM STUDENT)

1. Подсчитать количество студентов в каждом из университетов . 1 sql-команда.

SELECT UNAME, COUNT (ID\_STUDENT) AS 'STUDENTS AMOUNT'

FROM STUDENT

JOIN UGROUP ON UGROUP.ID\_GROUP = STUDENT.ID\_GROUP

JOIN UNIVERSITY ON UNIVERSITY.ID\_UNIVERSITY = UGROUP.ID\_UNIVERSITY

GROUP BY UNAME

