**Лабораторная работа № 5. Использование оператора SELECT**

|  |  |
| --- | --- |
| Задание | Выполнение |
| 1**.** С помощью оператора SELECT сформировать результирующий набор, содержащий все строки и столбцы таблицы **FACULTY** базы данных **CREATE\_XXX\_ UNIVER**. | SELECT \* FROM FACULTY; |
| 2.С помощью оператора SELECT сформировать результирующий набор, содержащий все строки столбцов **TEACHER** и **PULPIT** таблицы **TEACHER**. | SELECT TEACHER, PULPIT FROM TEACHER; |
| 3.Сформировать перечень имен преподавателей (таблица **TEACHER**), работающих на кафедре **ИСиТ** (столбец **PULPIT**). | SELECT TEACHER\_NAME FROM TEACHER  WHERE PULPIT = 'ИСиТ'; |
| 4.Сформировать перечень имен (столбец **TEACHER\_NAME**) преподавателей (таблица **TEACHER**), работающих на кафедре **ИСиТ** или на кафедре **ПОиСОИ** (столбец **PULPIT**). | SELECT TEACHER\_NAME FROM TEACHER  WHERE PULPIT = 'ИСиТ' OR PULPIT = 'ПОиСОИ'; |
| 5.Сформировать перечень имен (столбец **TEACHER\_NAME**) преподавателей (таблица **TEACHER**) женского пола (столбец **GENDER**), работающих на кафедре **ИСиТ**. | SELECT TEACHER\_NAME FROM TEACHER  WHERE PULPIT = 'ИСиТ' AND GENDER = 'ж'; |
| 6.Сформировать перечень имен (столбец **TEACHER\_NAME**) всех преподавателей (таблица **TEACHER**) кроме преподавателей женского пола (столбец **GENDER**), работающих на кафедре **ИСиТ**. При этом столбец результирующего набора должен иметь имя: **Имя преподавателя**. | SELECT TEACHER\_NAME [Имя преподавалеля]  FROM TEACHER  WHERE PULPIT = 'ИСиТ' AND GENDER !='ж'; |
| **7.** На основе содержимого таблицы **TEACHER** сформировать перечень кодов всех кафедр. | SELECT DISTINCT PULPIT FROM TEACHER; |
| 8.Сформировать перечень аудиторий (таблица **AUDITORIUM**) в порядке возрастания их вместимости (столбец **AUDITORIUM\_CAPACITY**). Примечание: использовать секцию ORDER BY. | SELECT AUDITORIUM, AUDITORIUM\_CAPACITY FROM AUDITORIUM  ORDER BY AUDITORIUM\_CAPACITY; |
| 9. На основе содержимого таблицы **AUDITORIUM** выбрать два типа аудиторий (столбец **AUDITORIUM\_TYPE**), к которым относятся самые вместительные аудитории. Примечание: использовать секции ORDER BY, ключевые слова DISTINCT и TOP. | SELECT DISTINCT TOP(2) AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY  FROM AUDITORIUM  ORDER BY AUDITORIUM\_CAPACITY DESC; |
| 10.На основе содержимого таблицы **PROGRESS** сформировать перечень дисциплин (столбец **SUBJECT**), по которым имеются оценки в пределах от 8 до 10. Примечание: использовать ключевое слово DISTINCT и предикат BETWEEN. | SELECT DISTINCT SUBJECT FROM PROGRESS  WHERE NOTE BETWEEN 8 AND 10; |
| 11.На основе содержимого таблицы **SUBJECT** сформировать перечень дисциплин (столбец **SUBJECT**), которые закреплены за кафедрами (столбец **PULPIT**) **ЛЗиДВ**, **ПОиСОИ** и **ОВ**. Примечание: использовать секцию DISTINCT и ключевое слово IN. | SELECT DISTINCT SUBJECT FROM SUBJECT  WHERE PULPIT IN ('ЛЗиДВ','ПОиСОИ','ОВ'); |
| 12. На основе содержимого таблицы **PROFESSION** определить перечень наименований специальностей (столбец **PROFESSION\_NAME**), которым соответствует квалификация (столбец **QUALIFICATION**) в наименовании которого присутствует слово **химик**. Результирующий набор запроса должен содержать два столбца с именами: **Наименование специальности** и **Квалификация**. Примечание: использовать LIKE. | SELECT PROFESSION\_NAME [Наименование специальности],  QUALIFICATION [Квалификация]  FROM PROFESSION  WHERE QUALIFICATION LIKE '%химик%'; |
| 13. С помощью оператора **SELECT** перенести данные о студентах (ФИО и дата рождения) во временную таблицу | CREATE TABLE #FIO\_DATE (  NAME nvarchar(100),  BDAY date  );  INSERT INTO #FIO\_DATE (NAME, BDAY)  SELECT NAME, BDAY  FROM STUDENT;  SELECT \* FROM #FIO\_DATE; |
| 14. Разработать и выполнить аналогичные запросы для базы данных **X\_MyBASE**. | USE STASH\_MyBase;  SELECT \* FROM Предмет;  SELECT Название\_предмета FROM Предмет  WHERE Объём\_лекций BETWEEN 20 AND 40;  SELECT \* FROM Студент;  SELECT Телефон FROM Студент  WHERE Фамилия LIKE 'A%';  SELECT DISTINCT TOP(2) \* FROM Оценка;  CREATE TABLE #INFO  (  FAMILY nvarchar(50)  );  INSERT INTO #INFO(FAMILY)  SELECT Фамилия FROM Студент;  SELECT \* FROM #INFO; |
| 15\*. По таблице **PROGRESS** определить средний балл студентов, у которых нет неудовлетворительных оценок. | SELECT AVG(NOTE) FROM PROGRESS;  SELECT AVG(NOTE) FROM PROGRESS  WHERE NOTE > 5; |