**ПРАКТИКАЛЫҚ ЖҰМЫС №7. СҰРЫПТАУ МЕН ТОПТАУҒА АРНАЛҒАН СҰРАНЫСТАР ҚҰРУ. МӘЛІМЕТТЕРДІ ӨЗГЕРТУГЕ АРНАЛҒАН СҰРАНЫСТАР. ФУНКЦИЯЛАРДЫ ҚОЛДАНУ. Мақсаты**

SELECT опеарторындағы топтау және сұрыптау операторларын қолдануды үйрену. Сұраныстарды кіріктірілген функцияларды қолдану.

**7.2. Кіріс мәліметтер**

Жеке өзіндік жұмыс және 6 практикалық жұмыс нәтижесі.

**7.3. Қолданылатын программа**

SQL Server Managеmant Studio.

**7.4. Тапсырма**

**1. Кіріктірілген функцияларды қолданып, сұраныстар құру**

Жазылуы:

**Функция\_аты[(аргумент[, аргумент]...)]**

**Агрегаттық функциялар**

**AVG – орта мән**

**MIN - минимум**

**SUM** - қосынды

**COUNT** – санын есептеу

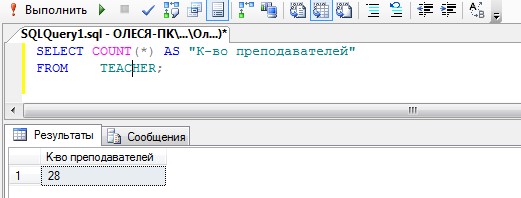
**STDEV –** орта мән

**MAX** - максимум

**Сұраныс 1.**

**SELECT COUNT(\*) AS "Оқытушылар саны"**

**FROM TEACHER;**



**Өз бетінше**  сұраныс 2. Телефоны жоқ ассистенттер санын анықтау

**Өз бетінше** сұраныс 3. Физика-математика факультетінде қанша кафедра барын анықтау

**Сұраныс 4. «Компьютерлік жүйелер және желілер» кафедрасының оқытушылары қанша қызметте жасайды?**

**SELECT C0UNT(DISTINCT DOLGNOST)**

**FROM KAFEDRA d, TEACHER t**

**WHERE d.KOD\_KAFEDRU = t.KOD\_KAFEDRU AND**

**LOWER(d.NAME\_KAFEDRU) = 'Компьютерлік жүйелер және желілер';**

**SUM функциясы**

**Сұраныс 5. Асситенттердің ставкаларының қосындысын табу**

**SELECT SUM(Salary)**

**FROM TEACHER**

**WHERE LOWER(DOLGNOST) = 'ассистент';**

**AVG функциясы**

**Өз бетінше** сұраныс 6. Барлық оқытушылардың ставкаларының орташа мәнін анықтау

**Өз бетінше** сұраныс 7. ЖОО-ндағы ставкалардың орташа мәнін анықтау. Cұраныста DISTINCT кілттік сөзін қолдану керек.

**MIN және MAX функциялары**

**Запрос 8.Оқытушының максимальды жалақысы барын анықтау**

**SELECT MAX(Salary + Rise)**

**FROM TEACHER;**

**Өз бетінше сұраныс** 9. Информатика кафедрасына ең соңғы жұмысқа қабылданған оқытушыны анықтау(жұмысқа алынған максимальды дата)

**Агрегатты функцияларды қолданып өрнектер құрастыру**

**Сұраныс 10.Ставкалар қосындысының жалақылар қосындысына пайздық қатынасын шығару және керісінше.**

**SELECT SUM(Salary)\*100/SUM(Rise) AS "Жалақының жалақыға пайыздық қатынасы",**

**SUM(Rise)\*100/SUM(Salary) AS "Жалақының ставкаға пайыздық қатынасы"**

**FROM TEACHER;**

**Жолдық функциялар**

**UPPER, LOWER функциялары**

**Сұраныс 11. Оқытушылардың фамилияларын үлкен әріптермен шығару**

**SELECT UPPER(NAME\_TEACHER) AS "Үлкен әріппен жазу”**

**FROM TEACHER;**

**Сандық функциялар**

ABS абсолют мән

DEGREES Возвращает для значения угла в радианах соответствующее значение в градусах.

RAND – float типті кездейсоқ сан от 0 мен 1 аралығында.

EXP экспонента

ROUND – көрсетілген ұзындық пен дәлдікте жуықталған сандық мәнді қайтарады.

FLOOR көрсетілген саннан үлкен болатын бүтін санға жуықтап, мәнді қайтарады.

LOG логорифм

SIN - синус

LOG10 ондық логорифм

SQRT – корень квадратный

PI - пи саны, 3.14

SQUARE – квадрат түбір

POWER – санның дәрежесін табу.

TAN - тангенс

**Уақытша функциялар**

**Дата мен уақыттан бөліп алу функциясы.**

**DATENAME ( datepart , date )**

Мұндағы datepart -мәндері: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND және т.б.

DATEPART (datepart,date) – датадан уақыт пен датаның мәндерін бөліп қайтарады.

DAY (date) - датаның күнін қайтарады.

MONTH (date) – датадан айды бөліп алады.

YEAR (date) – датадан жылды бөліп алады.

**Сұраныс 12. 1979 жылы туылған оқытушылардың фамилияларын шығару.**

**SELECT Name\_teacher, BIRTHDAY**

**FROM TEACHER**

**WHERE DATENAME(YEAR, BIRTHDAY)=1979;**

**Жүйелік дата мен уақыттың мәнін алатын функциялар**

GETDATE ( ) -ағымдағы датаны қайтарады.

**Датадан мәндерді бөлім алу**

Функция DATEADD (datepart, number , date ) - көрсетілген датаға datepart бөлігінде көрсетілген атауға сәйкес мән қосылған жаңа мәнді қайтарады.

Datepart мәндері: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND.

мысалы, DATEADD(month, 1, '2006-08-30')

**Сұраныс 13. SELECT NAME\_TEACHER, DATA\_HIRE AS ' Жұмысқа қабылданған күн',**

**DATEADD(month, 3, DATA\_HIRE) AS ' Плюс 3 ай '**

**FROM TEACHER**

**WHERE (NAME\_TEACHER) LIKE 'С%';**

**LAST\_DAY(дата) -айдың соңына дейін қанша күн қалғанын анықтау**

**DATEDIFF функциясы**

**DATEDIFF ( datepart , startdate , enddate ) – даталардың айырмасы datepart аргументінде көрсетілген мәнмен қайтарады.**

**Сұраныс 14. Қызметкердің жұмыс жасаған уақытын анықтау:**

**SELECT 'Арманның жұмыс жасаған уақыты ' ||**

**R0uND(DATEDIFF(month,GETDATE(), DATA\_HIRE),1) ||**

**' ай' AS "Арманның стажы"**

**FROM TEACHER**

**WHERE NAME\_TEACHER LIKE 'Арман%';**

**NEXT\_DAY функциясы**

**NEХТ\_DАУ(дата, день\_недели)**

**Типті ауыстыру функциялары**

**CAST және CONVERT функциялары**

**CAST ( expression AS data\_type [ ( length ) ] )**

**CONVERT ( data\_type [ ( length ) ] , expression [ , style ] )**

**Мысалы**

**CAST(10.3496847 AS money)**

**CAST(10.6496 AS int)**

**2. Топтау және сұрыптау**

**HAVING, GROUP BY және ORDER BY операторлары.**

**Топтау**

**Сұраныс 15. Әрбір корпус үшін орналасқан кафедралар санын анықтау.**

**SELECT NUM\_KORPUSA AS “Корпус”,**

**СOUNT(\*) AS "K-вo кафедр"**

**FROM KAFEDRA**

**GROUP BY NUM\_KORPUSA ;**

**Өз бетінше сұраныс** 16. Әрбір қызмет үшін жалақыларының қосындысын есептеу.

**Ө бетінше сұраныс** 17. 1 корпуста орналасқан әрбір факультет үшін топтар саны мен студенттер санын есептеу.

**Бірнеше баған бойынша топтау**

GROUP BY операторы.

**Сұраныс 18.** 1 корпуста орналасқан әрбір факультет үшін әр топтағы студенттер санын есептеу.

**SELECT f.Name\_faculteta,**

**s."GROUP", count(s."GROUP") AS "Кол-во студентов в группе" FROM FACULTET f, KAFEDRA d, STUDENT s**

**WHERE f.KOD\_FACULTETA = d.KOD\_FACULTETA AND d.KOD\_kafedru = s.KOD\_kafedru AND**

**d.NUM\_KORPUSA = '1'**

**GROUP BY f.Name\_faculteta,s."GROUP";**

**Өз бетінше сұраныс** 19. Әрбір кафедра және қызмет үшін оқытушыларды жалақыларының қосындысын және орта мәнін анықтау.

**Өрнектерді қолдану**

**Сұраныс 20**. Для каждого значения зарплаты, не превышающего 1500, вывести это значение и количество преподавателей, такую зарплату получающих. **SELECT Salary + Rise, C0UNT(\*)**

**FROM TEACHER**

**WHERE Salary + Rise <= 1500**

**GROUP BY Salary + Rise;**

**Іштестірілген агрегаттық функциялар**

**Сұраныс 21**. Бір қызметті жасайтын оқытушылардың минимальды және максимальды ставкаларының орташа мәнін анықтау, орташа ставкалардың минималь және максималь мәнерін анықтау.

**SELECT AVG(MIN(Salary)) AS AVG\_MIN,**

**AVG(MAX(Salary)) AS AVG\_MAX,**

**MIN(AVG(Salary)) AS MIN\_AVG,**

**MAX(AVG(Salary)) AS MAX\_AVG**

**FROM TEACHER**

**GROUP BY Dolgnost ;**

**Шартпен топтау**

**Сұраныс 22**. Профессорлар саны 1-ден жоғары болтаны кафедра номерін шығару.

**SELECT KOD\_kafedru as "Номер кафедры" ,Count(\*) as "Кол-во профессоров на кафедре"**

**FROM TEACHER**

**WHERE dolgnost='профессор' GROUP BY KOD\_kafedru**

**having count(dolgnost) > 1 ;**

**HAVING операторын қолдану**

**Сұраныс 23**. Бір немесе одан көп жұмыс жасайтын профессорлар бар физика-математика факультетінің кафедраларын шығару. Профессорлар санын және жалақыларының қосындысын көрсету.

**SELECT d.Name\_kafedru, Count(\*), SUM(t.salary + t.Rise)**

**FROM FACULTET f, KAFEDRA d, TEACHER t**

**WHERE f.KOD\_FACULTETA = d.KOD\_FACULTETA AND d.KOD\_kafedru = t.KOD\_kafedru AND**

**LOWER(f.Name\_faculteta) = 'математики и информатики' AND**

**LOWER(t.Dolgnost ) = 'профессор'**

**GROUP BY d.Name\_kafedru**

**HAVING COUNT(\*) > 0;**

**GROUP BY операторын қолданбай HAVING қолдану**

**Сұраныс 24**.

**SELECT MIN(Salary), MAX(Rise), SUM(Salary + Rise)**

**FROM TEACHER**

**HAVING SUM(Salary + Rise) > 15000;**

**Сұраныс 25.**

**SELECT AVG(Salary), AVG(Rise), SUM(Salary + Rise)**

**FROM TEACHER**

**WHERE LOWER(Dolgnost ) = 'ассистент'**

**HAVING SUM(Salary + Rise) > 2500;**

Практика жүзінде HAVING GROUP BY операторынсыз қолданылмайды, себебі көп МҚБЖ-нде мұндай мүмкіндік жоқ.

**Сұрыптау**

**Сұраныс 26.**

**SELECT NAME\_TEACHER FROM TEACHER wheRE LOwER(Dolgnost ) =’профессор’ OR**

**LOWER(Dolgnost ) = 'доцент'**

**ORDER BY NAME\_TEACHER;**

**Сұраныс 27.** Өрнек бойынша сұрыптауды орындау.

**SELECT Name\_teacher, Salary + Rise**

**FROM TEACHER**

**WHERE LOWER(Dolgnost ) = 'асcистент’**

**ORDER BY Salary + Rise;**

**Сұрыптау бағытын анықтау**

ASC (өсу ретімен)

DESC (кему ретімен).

**Сұраныс 28**.

**SELECT Name\_teacher, Data\_hire**

**FROM TEACHER**

**WHERE LOWER(Dolgnost) = 'ассистент' ORDER BY Data\_hire ASC;**

Өз бетінше сұраныс 29. Доцеттерді жалақысы бойынша кері ретпен шығару.

**№7 практикалық жұмыс тапсырмасы**

|  |
| --- |
| **Transact-SQL тілінде 14 сұраныс құру:**   * 1 сұраныс - **COUNT** ; * 1 сұраныс- **SUM**; * 1 сұраныс- **UPPER, LOWER**; * 1 сұраныс даталық функциялар; * 1 сұраныс -бір баған бойынша топтау; * 1 сұраныс-бірнеше баған бойынша топтау; * 1 сұраныс- **HAVING операторын қолдану**; * 1 сұраныс - **HAVING** операторы **GROUP BY қолданбай;** * 1 сұраныс-сұрыптау;     Сұраныстарды ***студентФИО/Лаб7****.doc файлына сақтау* |