**И .Ф .Аста хов а , А.П .Т олстобров , В.М .М ельников**

**SQL**

в п римерах и зад ач ах

У ч ебное п особие

Доп ущ ено Н ауч но-метод ич еским сов етом п о сп ециальности 010200 “Приклад ная математика и информатика” У М О унив ерситетов РФ

Воронеж 2001

С о д ер ж а н и е

**В В Е Д Е НИ Е ........................................................................................6**

1. **О С НО В НЫ Е П О НЯ Т И Я И О П РЕ ДЕ ЛЕ НИ Я .....................................9**
   1. О сн о вн ы е по н яти я р еляци о н н ы х б а з да н н ы х ............................9
   2. О тли чи е SQL о т пр о цедур н ы х языко в пр о гр а мми р о ва н и я .......12 1.3. Ин тер а кти вн ы й и встр о ен н ы й SQL..........................................12
   3. С о ста вн ые ча сти SQL.............................................................13
   4. Ти пы да н н ы х SQL...................................................................14
      1. Ти п да н н ы х “стр о ка си мво ло в” .........................................14 1.5.2. Ч и сло вы е ти пы да н н ы х ....................................................15 1.5.3. Да та и вр емя ...................................................................16

1.5.4. Н ео пр еделен н ы е и ли пр о пущен н ы е да н н ы е (NULL)..........17

* 1. Усло ви я и тер ми н о ло ги я..... **О ш ибка! З акл адка не определ ена.**
  2. Учеб н а я б а за да н н ы х .............................................................19

1. **В Ы БО РКА Д АННЫ Х (О П Е РАТ О Р** SELECT**)..................................22**
   1. Пр о стейши е SELECT за пр о сы ...............................................22
   2. О пер а то р ы IN, BETWEEN, LIKE, IS NULL.................................28
   3. Пр ео б р а зо ва н и е вы во да и встр о ен н ые фун кци и .....................31
      1. Ч и сло вы е, си мво льн ы е и стр о ко вы е ко н ста н ты ................31
      2. Ар и фмети чески е о пер а ци и для пр ео б р а зо ва н и я

чи сло вы х да н н ы х.............................................................32

* + 1. С и мво льн а я о пер а ци я ко н ка тен а ци и стр о к.......................32
    2. С и мво льн ы е фун кци и пр ео б р а зо ва н и я б укв р а зли чн ы х

сло в в стр о ке...................................................................33

* + 1. С и мво льн ы е стр о ко вые фун кци и ......................................33 2.3.6. Фун кци и р а б о ты с чи сла ми ...............................................36

2.3.7. Фун кци и пр ео б р а зо ва н и я зн а чен и й..................................38

* 1. Агр еги р о ва н и е и гр уппо вы е фун кци и ......................................42
  2. Пусты е зн а чен и я (NULL) в а гр еги р ующи х фун кци ях.................45
     1. Вли ян и е NULL–зн а чен и й в фун кци и COUNT.......................45
     2. Вли ян и е NULL–зн а чен и й в фун кци и AVG...........................46
  3. Р езульта т действи я тр ехзн а чн ы х усло вн ы х о пер а то р о в ..........46
  4. Упо р ядо чен и е вы хо дн ы х по лей (ORDER BY).............................48
  5. Вло ж ен н ы е по дза пр о сы ..........................................................50 2.9. Фо р ми р о ва н и е связа н н ы х по дза пр о со в...................................52 2.10. С вяза н н ые по дза пр о сы в HAVING..........................................53
  6. Испо льзо ва н и е о пер а то р а EXISTS........................................55
  7. О пер а то р ы ср а вн ен и я с мн о ж ество м зн а чен и й IN, ANY,

ALL.......................................................................................57

* 1. О со б ен н о сти о пер а то р о в ANY, ALL, EXISTS пр и

о б р а б о тке NULL ....................................................................59

* 1. Испо льзо ва н и е COUNT вместо EXISTS................................61 2.15. О пер а то р о б ъ еди н ен и я UNION............................................62 2.16. Устр а н ен и е дуб ли р о ва н и я в UNION.......................................63

2.17. Испо льзо ва н и е UNION с ORDER BY.........................................65 2.18. Вн ешн ее о б ъ еди н ен и е..........................................................66

2.19. С о еди н ен и е та б ли ц с и спо льзо ва н и ем о пер а то р а JOIN.........67

* + 1. О пер а ци и со еди н ен и я та б ли ц по ср едство м

ссы ло чн о й цело стн о сти ...................................................69

* + 1. Вн ешн ее со еди н ен и е та б ли ц .........................................70
    2. Испо льзо ва н и е псевдо н и мо в пр и со еди н ен и и та б ли ц.....73

1. **МАНИ П УЛ И РО В АНИ Е ДАННЫ МИ ...............................................77**
   1. Ко ма н ды ма н и пули р о ва н и я да н н ы ми ......................................77
   2. Испо льзо ва н и е по дза пр о со в в INSERT....................................81
      1. Испо льзо ва н и е по дза пр о со в, о сн о ва н н ы х н а та б ли ца х

вн ешн и х за пр о со в............................................................82

* + 1. Испо льзо ва н и е по дза пр о со в с DELETE.............................82
    2. Испо льзо ва н и е по дза пр о со в с UPDATE.............................84

1. **С О З Д АНИ Е О БЪЕ КТ О В БАЗ Ы Д АННЫ Х ....................................86**
   1. С о зда н и е та б ли ц б а зы да н н ы х ...............................................86
   2. Испо льзо ва н и е и н декса ци и для б ы стр о го до ступа к

да н н ы м .................................................................................87

* 1. Измен ен и е существующей та б ли цы ........................................88
  2. Уда лен и е та б ли цы .................................................................89
  3. О гр а н и чен и я н а мн о ж ество до пусти мы х зн а чен и й да н н ы х.......90
     1. О гр а н и чен и я NOT NULL....................................................91
     2. Ун и ка льн о сть ка к о гр а н и чен и е н а сто лб ец........................92 4.5.3. Ун и ка льн о сть ка к о гр а н и чен и е та б ли цы ...........................93 4.5.4. Пр и сво ен и е и мен о гр а н и чен и ям.......................................94 4.5.5. О гр а н и чен и е пер ви чн ы х ключей.......................................94 4.5.6. С о ста вн ы е пер ви чн ые ключи ...........................................95

4.5.7. Пр о вер ка зн а чен и й по лей................................................96 4.5.8. Пр о вер ка усло ви й с и спо льзо ва н и ем со ста вн ы х по лей.....96

4.5.9. Уста н о вка зн а чен и й по умо лча н и ю...................................97

* 1. По ддер ж ка цело стн о сти да н н ы х ...........................................100
     1. Вн ешн и е и р о ди тельски е ключи .....................................100 4.6.2. С о ста вн ы е вн ешн и е ключи .............................................101 4.6.3. С мы сл вн ешн его и р о ди тельско го ключей......................101 4.6.4. О гр а н и чен и е FOREIGN KEY (вн ешн его ключа )................102 4.6.5. Вн ешн и й ключ ка к о гр а н и чен и е та б ли цы ........................103
     2. Вн ешн и й ключ ка к о гр а н и чен и е сто лб цо в .......................104
     3. По ддер ж а н и е ссы ло чн о й цело стн о сти и о гр а н и чен и я

зн а чен и й р о ди тельско го ключа .......................................106

* + 1. Испо льзо ва н и е пер ви чн о го ключа в ка честве

ун и ка льн о го вн ешн его ключа ..........................................106

* + 1. О гр а н и чен и я зн а чен и й вн ешн его ключа .........................107
    2. Действи е о гр а н и чен и й вн ешн его и р о ди тельско го ключей пр и и спо льзо ва н и и ко ма н д мо ди фи ка ци и ...........107

1. **П РЕ ДС Т АВ Л Е НИ Я (**VIEW**)..........................................................111**
   1. Пр едста влен и я – и мен о ва н н ы е за пр о сы ...............................111
   2. Пр едста влен и я та б ли ц .........................................................112 5.3. Пр едста влен и я сто лб цо в......................................................112
   3. М о ди фи ци р о ва н и е пр едста влен и й.......................................113
   4. М а ски р ующи е пр едста влен и я ...............................................113
      1. Пр едста влен и я, ма ски р ующи е сто лб цы ..........................113
      2. О пер а ци и мо ди фи ка ци и в пр едста влен и ях,

ма ски р ующи х сто лб цы ...................................................114

* + 1. Пр едста влен и я, ма ски р ующи е стр о ки ............................114
    2. О пер а ци и мо ди фи ка ци и в пр едста влен и ях,

ма ски р ующи х стр о ки ......................................................115

* + 1. О пер а ци и мо ди фи ка ци и в пр едста влен и ях,

ма ски р ующи х стр о ки и сто лб цы ......................................116

* 1. Агр еги р о ва н н ы е пр едста влен и я............................................118
  2. Пр едста влен и я, о сн о ва н н ы е н а н еско льки х та б ли ца х ...........118
  3. Пр едста влен и я и по дза пр о сы ...............................................119
  4. О гр а н и чен и я пр и мен ен и я о пер а то р а SELECT для со зда н и я

пр едста влен и й....................................................................120

* 1. Уда лен и е пр едста влен и й....................................................120
  2. Измен ен и е зн а чен и й в пр едста влен и ях...............................121 5.12. Пр и мер ы о б н о вляемы х и н ео б н о вляемы х пр едста влен и й...122

5.13. Пр едста влен и я, б а зи р ующи еся н а др уги х пр едста влен и ях..124

1. **О П РЕ Д Е ЛЕ НИ Е П РАВ ДО С Т УП А П О Л ЬЗ О В АТ Е ЛЕ Й**

**К Д АННЫ М.................................................................................126**

* 1. По льзо ва тели и пр и ви леги и ..................................................126 6.2. С та н да р тн ые пр и ви леги и ......................................................127 6.3. Ко ма н да GRANT ....................................................................128
  2. Испо льзо ва н и е а р гумен то в ALL и PUBLIC.............................129
  3. О тмен а пр и ви леги й..............................................................130
  4. Испо льзо ва н и е пр едста влен и й для фи льтр а ци и

пр и ви леги й.........................................................................130

* + 1. О гр а н и чен и е пр и ви леги и SELECT для о пр еделен н ы х

сто лб цо в........................................................................131

* + 1. О гр а н и чен и е пр и ви леги й для о пр еделен н ы х стр о к.........131
    2. Пр едо ста влен и е до ступа то лько к и звлечен н ы м

да н н ы м..........................................................................132

* + 1. Испо льзо ва н и е пр едста влен и й в ка честве

а льтер н а ти вы к о гр а н и чен и ям........................................133

6.7. Др уги е ти пы пр и ви леги й.......................................................133 6.8. Ти пи чн ы е пр и ви леги и си стемы .............................................134

* 1. С о зда н и е и уда лен и е по льзо ва телей....................................135
  2. С о зда н и е си н о н и мо в (SYNONYM)..........................................136
  3. С и н о н и мы о б щего по льзо ва н и я (PUBLIC)..........................138
  4. Уда лен и е си н о н и мо в...........................................................138 **7.** **УП РАВ Л Е НИ Е Т РАНЗ АКЦИ Я МИ ................................................139** **П РИ Л О Ж Е НИ Е 1. О Т В ЕТ Ы К УП РАЖ ЕНЕ НИ Я М … ........................145**

**П РИ Л О Ж Е НИ Е 2. З АД АЧИ П О П РО Е КТ И РО В АНИ Ю БД**

**ЛИ Т Е РАТ УРА.............................................................................159**

**П РЕ Д МЕ Т НЫ Й УКАЗ АТ ЕЛ Ь**

Работа в ып олнена п ри сод ейств ии Российского фонд а Фунд аментальных исслед ов аний, грант № 99-01-00327.

Введен и е

Информационные системы, исп ользующ ие баз ы д анных, в настоящ ее в ремя п ред став ляют собой од ну из в ажнейших областей сов ременных комп ьютерных технологий. С этой сферой св яз ана большая ч асть сов ременного рынка п рограммных п род уктов . О д ной из общ их тенд енций в разв итии таких систем яв ляются п роцессы интеграции и станд артизации, затрагив ающ ие структуры д анных и сп особы их обработки и интерп ретации, системное и п риклад ное п рограммное обесп еч ение, сред ств а разработки в заимод ейств ия комп онентов баз д анных и т.д . Сов ременные системы уп рав ления базами д анных (СУ Б Д) основ аны на реляционной мод ели п ред став ления д анных – в большой степ ени благод аря п ростоте и ч еткости ее концеп туальных п онятий и строгого математич еского обоснов ания.

Н еотъемлемая и в ажная ч асть любой системы, в ключ ающ ей базы д анных – яз ыков ые сред ств а, п ред остав ляющ ие в оз можность п олуч ения д оступ а к д анным и осущ еств ления необход имых д ейств ий над сод ержимым д анных, оп ред еления их структур, сп особов исп ользов ания и интерп ретации. Я зык SQL п ояв ился в 70-е год ы как од но из таких сред ств . Его п рототип был разработан фирмой IBM и изв естен п од назв анием SEQUEL (Structured English QUEry Language). SQL в обрал в себя д остоинств а реляционной мод ели, в ч астности д остоинств а лежащ его в ее основ е математич еского ап п арата реляционной алгебры и реляционного исч исления, исп ользуя п ри этом срав нительно небольшое ч исло оп ераторов и относительно п ростой

синтаксис.

Б лагод аря св оим кач еств ам яз ык SQL стал – в нач але “д е-факто”, а з атем – и официально утв ержд енным в кач еств е станд арта яз ыком работы с реляционными баз ами д анных. Этот станд арт п од д ержив ается в семи в ед ущ ими миров ыми фирма ми, д ейств ующ ими в области технологии баз д анных. Исп ользов ание в ыразительного и эффектив ного станд артного языка п озв олило обесп еч ить в ысокую степ ень незав исимости разрабатыв аемых п риклад ных п рограммных систем от конкретного тип а исп ользуемой СУ Б Д, сущ еств енно п од нять уров ень и унификацию инструментальных сред ств разработки п риложений, работающ их с реляционными баз ами д анных.

Г ов оря о станд арте языка SQL, след ует заметить, ч то большинств о его коммерч еских реализаций имеют некоторые, большие или меньшие, отступ ления от станд арта. Это, конеч но, ухуд шает сов местимость систем, исп ользующ их раз лич ные “д иалекты” SQL. Н о, с д ругой стороны, п олез ные расширения реализаций языка относительно станд арта обесп еч ив ают разв итие яз ыка и со в ременем в ключ аются в нов ые ред акции станд арта. У ч итыв ая место, з анимаемое языком SQL в сов ременных информационных технологиях, его знание необход имо любому сп ециалисту, работающ ему в этой области.

Данное п особие п ред наз нач ено в п ерв ую оч еред ь д ля п реп од ав ателей и студ ентов , и ориентиров ано на обуч ение основ ам исп ользов ания языка SQL п ри п ров ед ении п рактич еских занятий п о уч ебным курсам, св язанным с изуч ением информационных систем, основ анных на базах д анных. В настоящ ее в ремя такие курсы в ход ят в уч ебные п ланы ряд а унив ерситетских сп ециальностей. С этой целью в п особии большое в нима ние уд елялось п од бору материала д ля п римеров , а также зад ач и уп ражнений, необход имых д ля п олуч ения п рактич еских нав ыков состав ления SQL-зап росов к базе д анных, а в оп ред елениях и п римерах п риоритет отд ав ался п ростоте и д оход ч ив ости материала, в озможно, с некоторым ущ ербом строгости его изложения. По этой п рич ине в п особие не в ошли особенности яз ыка, требующ ие д ля осв оения и исп ользов ания более глубоких знаний о функциониров ании сов ременных СУ Б Д и информационных систем на их основ е, из уч ение и исп ользов ание которых имеет смысл только п ри услов ии п олуч ения нав ыков п рактич еского исп ользов ания базов ых конструкций языка. При изложении материа ла ав торы п о в озможности старались, кроме сп ециально огов оренных случ аев , не отступ ать от станд арта яз ыка SQL.

В п риложении 1 п особия п рив ед ены отв еты на большинств о п рив ед енных в нем зад ач . Примеры и зад ач и п ротестиров аны с исп ользов анием СУ Б Д Oracle и отеч еств енной СУ Б Д ЛИН ТЕР. ЛИН ТЕР п ред став ляет собой п олномасштабный кросс-п латформенный SQL-серв ер, соотв етств ующ ий основ ным миров ым станд артам, п ред ъяв ляемым к системам такого класса. Для некоммерч еского исп ользов ания уч ебным зав ед ениям он п ред остав ляется бесп латно. Б олее п од робную информацию о системе можно п олуч ить на официальном web-сайте комп ании РЕЛЭК С п о ад ресу [www.relex.ru](http://www.relex.ru/).

В п риложении 2 п рив ед ены тексты д оп олнительных зад ач п о п роектиров анию баз д анных. Эти зад ач и могут исп ользов аться в кач еств е тем курсов ых работ и д ля самостоятельной работы студ ентов .

А в торы над еются, ч то п особие окажется п олезным не только п реп од ав ателям и студ ентам, но и д ругим ч итателям, з аинтересов анным в п олуч ении нач альных п рактич еских нав ыков исп ольз ов ания языка SQL.

1. О сн о вн ы е по н яти я и о пр еделен и я

**1.1. О сновны е поня тия рел я ц ионны х баз данны х**

О снов ой сов ременных систем, исп ользующ их базы д анных, яв ляется *р еляци о нна я* *м о д ель* *д а нны х*. В этой мод ели д анные, п ред став ляющ ие информацию о п ред метной области, организов аны в в ид е д в умерных таблиц, назыв аемых *о т но ш ени ям и .* Н а рис. 1 п рив ед ен п ример такой таблицыотношения и п оясняются основ ные термины реляционной мод ели.

а тр и б ута

Ко р теж

Рис. 1. Пример таблицы-отношения реляционной базы д анных

* *О т но ш ени е* – это таблица, п од обная п рив ед енной на рис. 1, и состоящ ая из строк и столбцов . Верхняя строка таблицы-отношения назыв ается

*за го ло вко м о т но ш ени я*. Термины *о т но ш ени е* и *т а бли ца* обыч но уп отребляются как синонимы, од нако в яз ыке SQL исп ользуется термин *т а бли ца* .

* Строки таблицы-отношения назыв аются *ко р т еж а м и* или *за пи сям и* . Столбцы назыв аются *а т р и бут а м и* . Термины – атрибут, столбец, колонка, п оле обыч но исп ольз уются как синонимы. К ажд ый атрибут имеет имя, которое д олжно быть уникальным в конкретной таблице-отношении, од нако в разных таблицах имена атрибутов могут сов п ад ать.
* К олич еств о кортежей в таблице-отношении назыв ается *ка р д и на льным чи сло м* отношения, а колич еств о атрибутов назыв ается *ст епенью* отношения.
* *Ключ*или *пер ви чны й* *ключ* отношения – это уникальный ид ентификатор строк (кортежей), то есть такой атрибут (набор атрибутов ), д ля которого в любой момент в ремени в отношении не сущ еств ует строк с од инаков ыми знач ениями этого атрибута (набора атрибутов ). Н а п рив ед енном рисунке таблицы яч ейка с именем ключ ев ого атрибута имеет нижнюю границу в в ид е д в ойной ч ерты.
* *Д о м ен* отношения – это сов окуп ность з нач ений, из которых могут в ыбираться знач ения конкретного атрибута. То есть, конкретный набор имеющ ихся в таблице знач ений атрибута в любой момент в ремени д олжен быть п од множеств ом множеств а з нач ений д омена, на котором оп ред елен этот атрибут.

В общ ем случ ае на од ном и том же д омене могут быть оп ред елены знач ения разных атрибутов . Важным яв ляется то, ч то д омены в в од ят огранич ения на оп ерации срав нения знач ений раз лич ных атрибутов . Эти огранич ения состоят в том, ч то корректным образ ом можно срав нив ать межд у собой только знач ения атрибутов , оп ред еленных на од ном и том же д омене.

О тношения реляционной базы д анных облад ают след ующ ими св ойств ами:

* в отношениях не д олжно быть кортежей-д убликатов .
* кортежи отношений неуп оряд оч ены.
* атрибуты отношений также неуп оряд оч ены.

Из этих св ойств отношения в ытекают след ующ ие в ажные след ств ия.

* Из уникальности кортежей след ует, ч то в отношении *всегд а* имеется атрибут или набор атрибутов , п озв оляющ их *и д ент и ф и ци р о ва т ь* кортеж, д ругими слов ами в отношении ***в сегда*** есть п ерв ич ный ключ .
* Из неуп оряд оч енности кортежей след ует, в о-п ерв ых, ч то в отношении не сущ еств ует д ругого сп особа ад ресации кортежей, кроме ад ресации *по ключу*, в о-в торых, ч то в отношении не сущ еств ует таких п онятий как п ерв ый кортеж, п ослед ний, п ред ыд ущ ий, след ующ ий и т.д .
* Из неуп оряд оч енности атрибутов след ует, ч то ед инств енным сп особом их ад ресации в з ап росах яв ляется исп ольз ов ание наименов ания атрибута.

О тносительно св ойств а реляционного отношения, касающ егося отсутств ия кортежей-д убликатов , след ует сд елать в ажное замеч ание. В этом п ункте SQL не п олностью соотв етств ует реляционной мод ели. А именно, в отношениях, яв ляющ ихся рез ультатами з ап росов , SQL *д о пуска ет* налич ие од инаков ых строк. Д ля их устранения в зап росе исп ользуется ключ ев ое слов о **DISTINCT** (см. ниже).

Информация в реляционных базах д анных, как п рав ило, хранится не в од ной таблице-отношении, а в нескольких. При соз д ании нескольких таблиц в заимосв язанной информации п ояв ляется в озможность в ып олнения более сложных оп ераций с д анными, то есть более сложной обработки д анных. Д ля работы со св яз анными д анными из нескольких таблиц в ажным яв ляется п онятие так назыв аемых *внеш ни х ключей.*

*Внеш ни м ключо м* таблицы назыв ается атрибут или набор атрибутов этой таблицы, кажд ое знач ение которых в текущ ем состоянии таблицы в сегд а сов п ад ает со з нач ением атрибутов , яв ляющ ихся ключ ом, в д ругой таблице. Внешние ключ и исп ользуются д ля св языв ания знач ений атрибутов из разных таблиц. С п омощ ью в нешних ключ ей обесп еч ив ается так наз ыв аемая ссылоч ная целостность базы д анных, то есть согласов анность д анных, оп исыв ающ их од ни и те же объекты, но хранящ ихся в разных таблицах.

* 1. **О тл ичие SQL от проц едурны х я зы ков**

**програм мирования**

Я зык SQL относится к классу неп роцед урных языков п рограммиров ания. В отлич ие от унив ерсальных п роцед урных языков , которые также могут быть исп ользов аны д ля работы с базами д анных, язык SQL ориентиров а н не на *за пи си* , а на *м но ж ест ва* .

Это означ ает след ующ ее. В кач еств е в ход ной информации д ля формулируемого на языке SQL зап роса к базе д анных исп ользуется *м но ж ест во ко р т еж ей-за пи сей* од ной или нескольких таблиц-отношений. В результате в ып олнения зап роса также образуется *м но ж ест во кор т еж ей* результирующ ей таблицы-отношения. Другими слов ами, в SQL результатом любой оп ерации над отношениями также яв ляется отношение. Зап рос SQL зад ает не п роцед уру, то есть п ослед ов ательность д ейств ий, необход имых д ля п олуч ения рез ультата, а услов ия, которым д олжны уд ов летв орять кортежи результирующ его отношения, сформулиров а нные в термина х в ход ного (или в ход ных) отношений.

* 1. **И нтерактивны й и встроенны й SQL**

Сущ еств уют и исп ольз уются д в е формы яз ыка SQL: *и нт ер а кт и вны й SQL* и *вст р о енны й SQL*.

*Инт ер а кт и вный SQL* исп ользуется д ля неп осред ств енного в в од а SQLзап росов п ользов ателем и п олуч ения результата в интерактив ном режиме.

*Вст р о енны й SQL* состоит из команд SQL, в строенных в нутрь п рограмм, которые обыч но нап исаны на некотором д ругом яз ыке (Паскаль, С, С++ и д р.). Это д елает п рограммы, нап исанные на таких яз ыках, более мощ ными, гибкими и эффектив ными, обесп еч ив ая их п рименение д ля работы с д анными, хранящ имися в реляционных баз ах. При этом, од нако, требуются д оп олнительные сред ств а обесп еч ения интерфейса SQL с языком, в который он в страив ается.

Данная книга п осв ящ ена интерактив ному SQL, п оэтому в ней не обсужд аются в оп росы п остроения интерфейса, п озв оляющ его св язать SQL с д ругими яз ыками п рограммиров ания.

**1.4. С оставны е части SQL**

И интерактив ный, и в строенный SQL п од разд еляются на след ующ ие состав ные ч асти.

*Язы к О пр ед елени я Д а нны х* – DDL(Data Definition Language), д ает в озможность созд ания, изменения и уд аления различ ных объектов базы д анных (таблиц, инд ексов , п ользов ателей, п рив илегий и т.д .).

К ч ислу д оп олнительных функций языка оп ред еления д анных DDL могут быть в ключ ены сред ств а оп ред еления огранич ений целостности д анных, оп ред еления п оряд ка структур хранения д анных, оп исания элементов физич еского уров ня хранения д анных.

*Язы к О бр а бо т ки Д а нны х* – DML (Data Manipulation Language), п ред остав ляет в озможность в ыборки информации из базы д анных и ее п реобразов ания.

Тем не менее, это не д в а различ ных яз ыка, а комп оненты ед иного SQL.

**1.5. Т ипы данны х SQL**

В яз ыке SQL имеются сред ств а, п озв оляющ ие д ля кажд ого атрибута указ ыв ать тип д анных, которому д олжны соотв етств ов ать в се знач ения этого атрибута.

След ует отметить, ч то оп ред еление тип ов д анных яв ляется той ч астью, в которой коммерч еские реализ ации языка не п олностью согласуются с требов аниями официального станд арта SQL. Это объясняется, в ч астности, желанием сд елать SQL сов местимым с д ругими языками п рограммиров ания.

### 1.5.1. Т ип данны х “ст рока сим вол ов”

Станд арт п од д ержив ает только од ин тип д ля п ред став ления текста: **CHARACTER(CHAR)**. Этот тип д анных п ред став ляет собой симв ольные строки фиксиров анной д лины. Его синтаксис имеет в ид : **CHARACTER** [(*д ли на* )] или **CHAR** [(*д ли на* )].

Текстов ые знач ения п оля таблицы, д ля которого оп ред елен тип **CHAR**, имеют *ф и кси р ова нную* д лину, которая оп ред еляется п араметром *д ли на* . Этот п араметр может п ринимать з нач ения от 1 до 255, то есть строка может сод ержать д о 255 симв олов . Если в о в в од имой в п оле текстов ой константе фактич еское ч исло симв олов меньше ч исла, оп ред еленного п араметром *д ли на* , то эта константа ав томатич ески д оп олняется сп рав а п робелами д о зад анного ч исла симв олов .

Н екоторые реализ ации языка SQL п од д ержив ают в кач еств е тип а д анных строки п еременной д лины. Этот тип может обознач аться ключ ев ыми слов ами **VARCHAR()**, **CHARACTER** **VARYING** или **CHAR** **VARYING()**. О н оп исыв ает текстов ую строку, которая может иметь *пр о и зво льную* д лину д о оп ред еленного конкретной реализацией SQL максимума (в Oracle д о 2000 симв олов ). В отлич ие от тип а **CHAR** в этом случ ае п ри в в од е текстов ой константы, фактич еская д лина которой меньше з ад анной, не п роизв од ится ее д оп олнения п робелами д о зад анного максима льного знач ения.

К онстанты, имеющ ие тип **CHARACTER** и **VARCHAR**, в в ыражениях SQL заключ аются в од иноч ные кав ыч ки, на п ример ‘*т екст* ’.

След ующ ие п ред ложения экв ив алентны:

**VARCHAR**[(*д ли на* )], **CHAR** **VARYING**[(*д ли на* )],

**CHARACTER** **VARYING**[(*д ли на* )]

Если д лина строки не указана яв но, она п олагается рав ной од ному симв олу в о в сех случ аях.

По срав нению с тип ом **CHAR** тип д анных **VARCHAR** п озв оляет более экономно исп ользов ать п амять, в ыд еляемую д ля хранения текстов ых знач ений, и оказыв ается более уд обным п ри в ып олнении оп ераций св яз анных со срав нением текстов ых констант.

### 1.5.2. Числ овы е т ипы данны х

Станд артными ч ислов ыми тип ами д анных SQL яв ляются:

* **INTEGER** – исп ольз уется д ля п ред став ления целых ч исел в д иап азоне от

–231 д о +231.

* **SMOLLINT** – исп ользуется д ля п ред став ления целых ч исел в д иап азоне меньшем, ч ем д ля **INTEGER**, а именно – от –215 д о +215.
* **DECIMAL** (*т о чно ст ь*[,*м а сш т а б*]) *–* десятич ное ч исло с фиксиров анной точ кой, точ ность указыв ает, сколько знач ащ их цифр имеет ч исло. Масштаб указ ыв ает максимальное ч исло цифр сп рав а от точ ки
* **NUMERIC** (*т о чно ст ь*[,*м а сш т а б*]) *–* десятич ное ч исло с фиксиров анной точ кой, такое же, как и **DECIMAL**.
* **FLOAT** [(*т о чно ст ь*)] *–* ч исло с п лав ающ ей точ кой и указ анной минимальной точ ностью.
* **REAL** – ч исло такоеже, как п ри тип е **FLOAT**, за исключ ением того, ч то точ ность устанав лив ается п о умолч анию в зав исимости от конкретной реализации SQL**.**
* **DOUBLE PRECISION** – ч исло такоеже, как и **REAL**, но точ ность в д в а раз а п рев ышает точ ность д ля **REAL.**

СУ Б Д Oracle исп ольз ует д оп олнительно тип д анных **NUMBER** д ля п ред став ления в сех ч ислов ых д анных, целых, с фиксиров анной или п лав а ющ ей точ кой. Его синтаксис:

**NUMBER** [(*т о чно ст ь*[, *м а сш т а б*])]*.*

Если знач ение п араметра *т о чно ст ь* не указано яв но, оно п олагается рав ным 38. Знач ение п араметра *м а сш т а б* п о умолч анию п ред п олагается рав ным 0. Знач ение п араметра *т о чно ст ь* может изменяться от 1 д о 38; знач ение п араметра *м а сш т а б* может изменяться от –84 до 128. Исп ольз ов ание отрицательных знач ений масштаба означ ает сд в иг д есятич ной точ ки в сторону старших раз ряд ов . Н ап ример, оп ред еление **NUMBER** (7, –3) означ ает округление д о тысяч .

Тип ы **DECIMAL** и **NUMERIC** п олностью экв ив алентны тип у **NUMBER**.

Синтаксис: **DECIMAL** [*(т о чно ст ь*[, *м а сш т а б*])], **DEC** [(*т о чно ст ь*[, *м а сш т а б*])],

**NUMERIC** [(*т о чно ст ь*[, *м а сш т а б*])].

### 1.5.3. Дат а и врем я

Тип д анных, п ред назнач енный д ля п ред став ления *д а т ы* и *вр ем ени* , также яв ляется нестанд артным, хотя и ч резв ыч айно п олезным. Поэтому д ля точ ного в ыяснения того, какие тип ы д анных п од д ержив ает конкретная СУ Б Д, след ует обращ аться к ее д окументации.

В СУ Б Д Oracle имеется тип **DATE**, исп ольз уемый д ля хранения д аты и в ремени. Под д ержив аются д аты, нач иная от 1 янв аря 4712 г. д о н.э. и д о 31 д екабря 4712 г. При оп ред елении д аты без уточ нения в ремени п о умолч анию п ринимается в ремя п олуноч и.

Н алич ие тип а д анных д ля хранения д аты и в ремени п озв оляет п од д ержив ать сп ециальную арифметику д ат и в ремен. Добав ление к п еременной тип а **DATE** целого ч исла оз нач ает ув елич ение д аты на соотв етств ующ ее ч исло д ней, а в ыч итание соотв етств ует оп ред елению более ранней д аты.

К онстанты тип а **DATE** зап исыв аются в зав исимости от формата, п ринятого в оп ерационной системе. Н ап ример ‘03.05.1999’ или ‘12/06/1989’, или ‘03-nov-1999’, или ‘03-apr-99’.

### 1.5.4. Неопредел енны е ил и пропущ енны е данны е (NULL)

Для обознач ения отсутств ующ их, п роп ущ енных или неиз в естных знач ений атрибута в SQL исп ольз уется ключ ев ое слов о **NULL**. Дов ольно ч асто можно в стретить слов осоч етание “*а т р и бут и м еет зна чени е* **NULL**”. Строго гов оря, **NULL** не яв ляется з нач ением в обыч ном п онимании, а исп ользуется именно д ля обознач ения того факта, ч то д ейств ительное знач ение атрибута на са мом д еле п роп ущ ено или неизв естно. Это п рив од ит к ряд у особенностей, ч то след ует уч итыв ать п ри исп ользов ании знач ений атрибутов , которые могут наход иться в состоянии **NULL**.

* В агрегирующ их функциях, п оз в оляющ их п олуч ать св од ную информацию п о множеств у з нач ений атрибута, нап ример, суммарное или сред нее з нач ение, д ля обесп еч ения точ ности и од нознач ности толков ания результатов отсутств ующ ие или **NULL**-знач ения атрибутов игнорируются.
* У слов ные оп ераторы от булев ой д в узнач ной логики **true/false** расширяются д о трехз нач ной логики **true/false/unknown**.
* Все оп ераторы, з а исключ ением оп ератора конкатенации строк “ || ”, в озв ращ ают п устое з нач ение (**NULL**), если знач ение любого из оп еранд ов отсутств ует (имеет “з нач ение **NULL**”).
* Для п ров ерки на п устое знач ение след ует исп ольз ов ать оп ераторы **IS NULL** и **IS NOT NULL** (исп ользов ание д ля этого оп ератора срав нения “ = ” яв ляется ошибкой).
* Функции п реобразов ания тип ов , имеющ ие **NULL** в кач еств е аргумента, в озв ращ ают п устое з нач ение (**NULL)**.

**1.6. И спол ь зуем ы е т ерм ины и обозначения**

*Ключевые сло ва* – это исп ольз уемые в в ыражениях SQL слов а, имеющ ие сп ециальное наз нач ение (нап ример, они могут обознач ать конкретные команд ы SQL). К люч ев ые слов а нельзя исп ользов ать д ля д ругих целей, к п римеру, в кач еств е имен объектов базы д анных. В книге они в ыд еляются шрифтом: **КЛЮЧЕВОЕСЛОВО**.

*Ко м а нд ы* , или *пр ед ло ж ени я*, яв ляются инструкциями, с п омощ ью которых SQL обращ ается к баз е д анных. К оманд ы состоят из од ной или более логич еских ч астей, назыв аемых п ред ложениями. Пред ложения нач инаются ключ ев ым слов ом и состоят из ключ ев ых слов и аргументов .

О бъекты базы д а нных, имеющ ие имена (таблицы, атрибуты и д р.), в книге также в ыд еляются особым образом: ТАБЛИЦА1, АТРИБУТ\_2.

В оп исании синтаксиса команд SQL оп ератороп ред еления **“::=”** разд еляет оп ред еляемый элемент (слев а от оп ератора) и собств енно его оп ред еление (сп рав а от оп ератора); кв ад ратные скобки “[ ]” указыв ают *нео бяза т ельны й* элемент синтаксич еской конструкции; многоточ ие “...” указ ыв ает, ч то в ыражение, п ред шеств ующ ее ему, может п ов торяться любое ч исло раз ; фигурные скобки “**{ }”** объед иняют п ослед ов ательность элементов в *ло ги ческую гр уппу*, од ин из элементов которой д олжно быть обязательно исп ользов ан; в ертикальна я ч ерта “|” указыв ает, ч то ч асть оп ред еления, след ующ ая за этим симв олом, яв ляется од ним из в озможных в ариантов ; в углов ые скобки “< >” заключ аются элементы, которые объясняются п о мере того, как в в од ятся.

**1.7. Учебная база данны х**

В п рив од имых в п особии п римерах п остроения SQL-з ап росов и контрольных уп ражнениеях исп ольз уется база д анных, состоящ ая из след ующ их таблиц.

Таблица 1.1. STUDENT (Студент)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STUDENT\_ID | SURNAME | NAME | STIPEND | KURS | CITY | BIRTHDAY | UNIV\_ID |
| 1 | Ив анов | Ив ан | 150 | 1 | О рел | 3/12/1982 | 10 |
| 3 | Петров | Петр | 200 | 3 | К урск | 1/12/1980 | 10 |
| 6 | Сид оров | Вадим | 150 | 4 | Москв а | 7/06/1979 | 22 |
| 10 | К узнецов | Б орис | 0 | 2 | Б рянск | 8/12/1981 | 10 |
| 12 | Зайцева | О льга | 250 | 2 | Л ипецк | 1/05/1981 | 10 |
| 265 | Пав лов | А ндрей | 0 | 3 | Воронеж | 5/11/1979 | 10 |
| 32 | К отов | Пав ел | 150 | 5 | Б елгород | **NULL** | 14 |
| 654 | Л укин | А ртем | 200 | 3 | Воронеж | 1/12/1981 | 10 |
| 276 | Петров | А нтон | 200 | 4 | **NULL** | 5/08/1981 | 22 |
| 55 | Б елкин | Вадим | 250 | 5 | Воронеж | 7/01/1980 | 10 |
| … … | … … | … … | … … | … … | … … | … … | … … |

STUDENT\_ID – числовой код, идентифицирующий студента,

SURNAME – фамилия студента,

NAME – имя студента,

STIPEND – стипендия, которую получает студент,

KURS – курс, на котором учится студент,

CITY – город, в котором живет студент,

BIRTHDAY – дата рождения студента,

UNIV\_ID – числовой код, идентифицирующий университет, в котором учится студент.

Таблица 1.2. LECTURER (Преподаватель)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LECTURER\_ID | SURNAME | NAME | CITY | UNIV\_ID |
| 24 | К олесников | Б орис | Воронеж | 10 |
| 46 | Н иконов | Ив ан | Воронеж | 10 |
| 74 | Л агутин | Пав ел | Москва | 22 |
| 108 | Струков | Н иколай | Москва | 22 |
| 276 | Н иколаев | Виктор | Воронеж | 10 |
| 328 | Сорокин | А ндрей | О рел | 10 |
| … … | … … | … … | … … | … … |

LECTURER\_ID – числовой код, идентифицирующий преподавателя,

SURNAME – фамилия преподавателя,

NAME – имя преподавателя,

CITY – город, в котором живет преподаватель,

UNIV\_ID – идентификатор университета, в котором работает преподаватель.

Таблица 1.3. SUBJECT (Предмет обучения)

|  |  |  |  |
| --- | --- | --- | --- |
| SUBJ\_ID | SUBJ\_NAME | HOUR | SEMESTER |
| 10 | Информатика | 56 | 1 |
| 22 | Физ ика | 34 | 1 |
| 43 | Математика | 56 | 2 |
| 56 | История | 34 | 4 |
| 94 | А нглийский | 56 | 3 |
| 73 | Физ культура | 34 | 5 |
| … … | … … | … … | … … |

SUBJ\_ID – идентификатор предмета обучения,

SUBJ\_NAME – наименование предмета обучения,

HOUR – количество часов, отводимых на изучение предмета, SEMESTER – семестр, в котором изучается данный предмет.

Таблица 1.4. UNIVERSITY (Университеты)

|  |  |  |  |
| --- | --- | --- | --- |
| UNIV\_ID | UNIV\_NAME | RATING | CITY |
| 22 | МГ У | 606 | Москва |
| 10 | ВГ У | 296 | Воронеж |
| 11 | Н Г У | 345 | Н ов осибирск |
| 32 | РГ У | 416 | Ростов |
| 14 | Б Г У | 326 | Б елгород |
| 15 | ТГ У | 368 | Томск |
| 18 | ВГ МА | 327 | Воронеж |
| … … | … … | … … | … … |

UNIV\_ID – идентификатор университета,

UNIV\_NAME – название университета,

RATING – рейтинг университета,

CITY – город, в котором расположен университет.

Таблица 1.5. EXAM\_MARKS (Экзаменационные оценки)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EXAM\_ID | STUDENT\_ID | SUBJ\_ID | MARK |  | EXAM\_DATE |
| 145 | 12 | 10 | 5 |  | 12/01/2000 |
| 34 | 32 | 10 | 4 |  | 23/01/2000 |
| 75 | 55 | 10 | 5 |  | 05/01/2000 |
| 238 | 12 | 22 | 3 |  | 17/06/1999 |
| 639 | 55 | 22 | **NULL** |  | 22/06/1999 |
| 43 | 6 | 22 | 4 |  | 18/01/2000 |
| … … | … … | … … | … … |  | … … |

EXAM\_ID – идентификатор экзамена,

STUDENT\_ID – идентификатор студента,

SUBJ\_ID – идентификатор предмета обучения, MARK – экзаменационная оценка, EXAM\_DATE – дата экзамена.

Таблица 1.6. SUBJ\_LECT (Учебные дисциплины преподавателей)

|  |  |
| --- | --- |
| LECTURER \_ID | SUBJ\_ID |
| 24 | 24 |
| 46 | 46 |
| 74 | 74 |
| 108 | 108 |
| 276 | 276 |
| 328 | 328 |
| … … | … … |

LECTURER\_ID – идентификатор преподавателя, SUBJ\_ID – идентификатор предмета обучения.

### В О П РО С Ы

1. К акие п оля п рив ед енных таблиц яв ляются п ерв ич ными ключ ами?
2. К акие д анные хранятся в столбце 2 в таблице “Пред мет обуч ения”?
3. К ак п о-д ругому назыв ается строка? Столбец?
4. Поч ему мы не можем з ап росить д ля п росмотра п ерв ые п ять строк?
5. Вы б о р ка да н н ы х (о пер а то р **SELECT**)

**2.1. П рост ей ш ие SELECT-запросы**

О п ератор **SELECT** (ВЫБРАТЬ) языка SQL яв ляется самым в ажным и самым ч асто исп ольз уемым оп ератором. О н п ред наз нач ен д ля *вы бо р ки* информации из таблиц базы д анных. У п рощ енный синтаксис оп ератора **SELECT** в ыгляд ит след ующ им образом.

**SELECT [DISTINCT]** <спи со к а тр и б уто в>

**FROM** <спи со к та б ли ц>

**[WHERE** <усло ви е вы б о р ки>**]**

**[ORDER BY** <спи со к а тр и б уто в>**]**

**[GROUP BY** <спи со к а тр и б уто в>**]**

**[HAVING** <усло ви е>**]**

**[UNION** <вы р а ж ен и е с о пер а то р о м **SELECT**>**]**;

В кв ад ратных скобках указаны элементы, которые могут отсутств ов ать в зап росе.

К люч ев ое слов о **SELECT** сообщ ает базе д анных, ч то д анное п ред ложение яв ляется з ап росом *на и звлечени е* информации. После слов а **SELECT** ч ерез зап ятую п ереч исляются *на и м ено ва ни я по лей* (сп исок атрибутов ), сод ержимое которых з ап рашив ается.

О бяз ательным ключ ев ым слов ом в п ред ложении-зап росе **SELECT** яв ляется слов о **FROM** (ИЗ). За ключ ев ым слов ом **FROM** указыв ается сп исок разд еленных зап ятыми имен таблиц, из которых изв лекается информация.

Н ап ример,

**SELECT** NAME, SURNAME

**FROM** STUDENT;

Л юбой SQL-зап рос д олжен заканч ив аться симв олом “;” (*т о чка с за пят ой)*.

Прив ед енный з ап рос осущ еств ляет в ыборку в сех знач ений п олей NAME и SURNAME из таблицы STUDENT.

Его рез ультатом яв ляется таблица след ующ его в ид а:

|  |  |
| --- | --- |
| NAME | SURNAME |
| Ив ан | Ив анов |
| Петр | Петров |
| Вадим | Сидоров |
| Б орис | К узнецов |
| О льга | Зайцева |
| А ндрей | Пав лов |
| Пав ел | К отов |
| А ртем | Л укин |
| А нтон | Петров |
| Вадим | Б елкин |
| … … | … … |

Поряд ок след ов ания столбцов в этой таблице соотв етств ует п оряд ку п олей NAME и SURNAME, указанному в з ап росе, а не их п оряд ку в о в ход ной таблице STUDENT.

Если необход имо в ыв ести знач ения *всех* столбцов таблицы, то можно в место п ереч исления их имен исп ользов ать симв ол “ \* ” (зв езд оч ка).

**SELECT** \*

**FROM** STUDENT;

В д анном случ ае в рез ультате в ып олнения зап роса буд ет п олуч ена в ся таблица STUDENT.

Ещ е раз обратим в нимание на то, ч то п олуч аемые в рез ультате SQLзап роса таблицы не в п олной мере отв еч ают оп ред елению реляционного отношения. В ч астности, в них могут оказаться кортежи с од инаков ыми знач ениями атрибутов .

Н ап ример, зап рос “Получ ить сп исок назв аний город ов , гд е п рожив ают студ енты, св ед ения о которых наход ятся в таблице STUDENT”, можно зап исать в след ующ ем в ид е

**SELECT** CITY **FROM** STUDENT;

Его результатом буд ет таблица

|  |
| --- |
| CITY |
| О рел  К урск  Москв а  Б рянск  Л ипецк  **Воронеж**  Б елгород  **Воронеж**  **NULL**  **Воронеж**  … … |

Вид но, ч то в таблице в стреч аются од инаков ые строки (в ыд елены жирным шрифтом).

Для исключ ения из рез ультата **SELECT**-зап роса п ов торяющ ихся зап исей исп ользуется ключ ев ое слов о **DISTINCT** (ОТЛИЧНЫЙ). Если зап рос **SELECT** изв лекает множеств о п олей, то **DISTINCT** *и сключа ет* д убликаты строк, в которых знач ения *всех* в ыбранных п олей ид ентич ны.

Зап рос “О п ред елить сп исок наз в аний *р а зли чны х* городов , гд е п рожив ают студ енты, св ед ения о которых наход ятся в таблице STUDENT”, можно зап исать в след ующ ем в ид е.

**SELECT** **DISTINCT** CITY **FROM** STUDENT;

В результате п олуч им таблицу, в которой д убликаты строк исключ ены.

|  |
| --- |
| CITY |
| О рел  К урск  Москв а  Б рянск  Л ипецк  Воронеж  Б елгород  **NULL**  … … |

К люч ев ое слов о **ALL** (ВСЕ), в отлич ие от **DISTINCT**, оказ ыв ает п ротив оп оложное д ейств ие, то есть п ри его исп ользов ании п ов торяющ иеся строки *включа ют ся* в состав в ыход ных д анных. Режим, зад ав аемый ключ ев ым слов ом **ALL**, д ейств ует п о умолч анию, п оэтому в реальных зап росах д ля этих целей оно п рактич ески не исп ользуется.

Исп ользов ание в оп ераторе **SELECT** п ред ложения, оп ред еляемого ключ ев ым слов ом **WHERE** (ГДЕ), п озв оляет зад ав ать в ыражение услов ия (п ред икат), п ринимающ ее знач ение *и с т и на* или *л о ж ь* д ля з нач ений п олей строк таблиц, к которым обращ ается оп ератор **SELECT**. Пред ложение **WHERE** оп ред еляет, ***к а к ие ст р ок и*** *ука за нны х т а бли ц д о лж ны быт ь вы бр а ны* . В таблицу, яв ляющ уюся рез ультатом зап роса, в ключ аются только те строки, д ля которых услов ие (п ред икат), указанное в п ред ложении **WHERE**, п ринимает з нач ение *и с т и на* .

**П ример.**

Н ап исать зап рос, в ып олняющ ий в ыборку имен (NAME) в сех студ ентов с фамилией (SURNAME) Петр о в, св ед ения о которых наход ятся в таблице STUDENT.

**SELECT** SURNAME, NAME

**FROM** STUDENT

**WHERE** SURNAME = ‘Петр о в’;

Рез ультатом этого зап роса буд ет таблица:

|  |  |
| --- | --- |
| SURNAME | NAME |
| Петров | Петр |
| Петров | А нтон |

В зад ав аемых в п ред ложении **WHERE** услов иях могут исп ользов аться оп ерации срав нения, оп ред еляемые след ующ ими оп ераторами: = (рав но), > (больше), < (меньше), >= (больше или ра в но), <= (меньше или рав но), <> (не рав но), а также логич еские оп ераторы **AND**, **OR** и **NOT**.

Н ап ример, з ап рос д ля п олуч ения *и м ен* и *ф а м и ли й* студ ентов , обуч ающ ихся на *т р ет ьем* курсе и п олуч ающ их стип енд ию (размер стип енд ии *бо льш е нуля*) буд ет в ыгляд еть таким образ ом:

**SELECT** NAME, SURNAME

**FROM** STUDENT

**WHERE** KURS = 3 **AND** STIPEND > 0;

Рез ультат в ып олнения этого з ап роса имеет в ид :

|  |  |
| --- | --- |
| SURNAME | NAME |
| Петров | Петр |
| Л укин | А ртем |

**УП РАЖ НЕ НИ Я**

1. Н ап ишите зап рос д ля в ыв од а ид ентификатора (номера) п ред мета обуч ения, его наименов ания, семестра, в котором он ч итается, и колич еств а отв од имых на него ч асов д ля в сех строк таблицы SUBJECT.
2. Н ап ишите зап рос, п оз в оляющ ий в ыв ести в се строки таблицы EXAM\_MARKS, в которых п ред мет обуч ения имеет номер (SUBJ\_ID), рав ный 12.
3. Н ап ишите з ап рос, в ыбирающ ий в се д анные из таблицы STUDENT, расп оложив столбцы таблицы в след ующ ем п оряд ке: KURS, SURNAME, NAME, STIPEND.
4. Н ап ишите зап рос **SELECT**, который в ып олняет в ыв од наименов аний п ред метов обуч ения (SUBJ\_NAME) и след ом за ним колич еств а ч асов (HOUR) д ля кажд ого п ред мета обуч ения (SUBJECT) в 4-м семестре (SEMESTR).
5. Н ап ишите з ап рос, п озв оляющ ий п олуч ить из таблицы EXAM\_MARKS з нач ения столбца MARK (экзамена ционная оценка) д ля в сех студ ентов , исключ ив из сп иска п ов торение од ина ков ых строк.
6. Н ап ишите зап рос, который в ып олняет в ыв од сп иска фамилий студ ентов , обуч ающ ихся на третьем и более старших курсах.
7. Н ап ишите з ап рос, в ыбирающ ий д анные о фамилии, имени и номере курса д ля студ ентов , п олуч ающ их стип енд ию больше 140.
8. Н ап ишите зап рос, в ып олняющ ий в ыборку из таблицы SUBJECT наз в аний в сех п ред метов обуч ения, на которые отв од ится более 30 ч асов .
9. Н ап ишите з ап рос, который в ып олняет в ыв од сп иска унив ерситетов , рейтинг которых п рев ышает 300 баллов .
10. Н ап ишите зап рос к таблице STUDENT д ля в ыв од а сп иска фамилий (SURNAME), имен (NAME) и номера курса (KURS) в сех студ ентов со стип енд ией большей или рав ной 100, и жив ущ их в Воронеже.
11. К акие д анные буд ут п олуч ены в результате в ып олнения зап роса?

**SELECT** \*

**FROM** STUDENT

**WHERE** (STIPEND < 100 **OR**

**NOT** (BIRTHDAY >= ‘10/03/1980’

**AND** STUDENT\_ID > 1003 ));

1. К акие д анные буд ут п олуч ены в результате в ып олнения зап роса?

**SELECT** \*

**FROM** STUDENT

**WHERE** **NOT** ((BIRTHDAY = ‘10/03/1980’ **OR** STIPEND > 100)

**AND** STUDENT\_ID > = 1003 );

**2.2. О ператоры IN,** **BETWEEN,** **LIKE,** **IS NULL**

При з ад ании логич еского услов ия в п ред ложении **WHERE** могут быть исп ользов аны оп ераторы **IN**, **BETWEEN**, **LIKE**, **IS** **NULL**.

О п ераторы **IN** (РАВЕН ЛЮБОМУ ИЗ СПИСКА) и **NOT** **IN** (НЕ РАВЕН НИ ОДНОМУ ИЗ СПИСКА) исп ольз уются д ля срав нения п ров еряемого знач ения п оля с зад анным сп иском. Этот сп исок знач ений указыв ается в скобках сп рав а от оп ератора **IN**.

Построенный с исп ользов анием **IN** п ред икат (услов ие) сч итается истинным, если з нач ение п оля, имя которого указано слев а от **IN**, *со впа д а ет* (п од разумев ается точ ное сов п ад ение) с од ним из знач ений, п ереч исленных в сп иске, указ анном в скобках сп рав а от **IN**.

Пред икат, п остроенный с исп ольз ов анием **NOT** **IN**, сч итается истинным, если знач ение п оля, имя которого указано слев а от **NOT** **IN**, *не со впа д а ет* ни с од ним из знач ений, п ереч исленных в сп иске, указанном в скобках сп рав а от **NOT** **IN**.

**П римеры.**

Получ ить из таблицы EXAM\_MARKS св ед ения о студ ентах, *и м еющи х* экзаменационные оценки только 4 и 5.

**SELECT** \*

**FROM** EXAM\_MARKS

**WHERE** MARK **IN** (4, 5);

Получ ить св ед ения о студ ентах, *не и м еющи х* ни од ной экзаменационной оценки, рав ной 4 и 5.

**SELECT** \*

**FROM** EXAM\_MARKS

**WHERE** MARK **NOT** **IN** (4, 5);

О п ератор **BETWEEN** исп ользуется д ля п ров ерки услов ия в хожд ения знач ения п оля в з ад анный интерв ал, то есть в место сп иска з нач ений атрибута этот оп ератор зад ает границы его из менения.

Н ап ример, зап рос, в ып олняющ ий в ыв од зап исей о п ред метах обуч ения, колич еств о ч асов , отв од имых на которые, лежит в п ред елах межд у 30 и 40, имеет в ид :

**SELECT** \*

**FROM** SUBJECT

**WHERE** HOUR **BETWEEN 30 AND 40**;

Г ранич ные знач ения, в д анном случ ае знач ения 30 и 40, *вход ят* в о множеств о знач ений, с которыми п роиз в од ится срав нение. О п ератор **BETWEEN** может исп ользов аться как д ля ч ислов ых, так и д ля симв ольных тип ов п олей.

О п ератор **LIKE** п рименим только к симв ольным п олям тип а **CHAR** или **VARCHAR** (см. раз д ел 1.5, Тип ы д анных SQL). Этот оп ератор осущ еств ляет п росмотр строков ых з нач ений п олей с целью оп ред еления, в ход ит ли зад анная в оп ераторе **LIKE** п од строка (образец п оиска) в симв ольную строку, яв ляющ уюся знач ением п ров еряемого п оля.

Для того, ч тобы осущ еств лять в ыборку строков ых знач ений п о зад анному образцу п од строки, можно п рименять шаблон искомого образца строки, исп ользующ ий след ующ ие симв олы:

* симв ол п од ч еркив ания “**\_**”, указ анный в шаблоне образ ца, оп ред еляет в оз можность налич ия в указанном месте *о д но го* *любого* симв ола,
* симв ол “%” д оп ускает п рисутств ие в указанном месте п ров еряемой строки п ослед ов ательности любых симв олов п роиз в ольной д лины.

**П ример.**

Н ап исать зап рос, в ыбирающ ий из таблицы **STUDENT** св ед ения о студ ентах, у которых фамилии нач инаются на букв у “Р”.

**SELECT** \*

**FROM** STUDENT

**WHERE** SURNAME **LIKE** ‘Р %’;

В случ ае в озникнов ения необход имости в ключ ения в образ ец д ля срав нения самих симв олов “**\_**” и “%” п рименяют, так назыв аемые escapeсимв олы. Если escape-симв ол п ред шеств ует знаку “**\_**” и “%”, то эти з наки буд ут интерп ретиров аться букв ально. Н ап ример, можно зад ать образец п оиска с п омощ ью след ующ его в ыражения **LIKE** ‘\_\\_Р ’ **ESCAPE** ‘\’.

В этом в ыражении симв ол ‘\’ с п омощ ью ключ ев ого слов а **ESCAPE** объяв ляется escape-симв олом. Перв ый симв ол “**\_**” в зад анном шаблоне п оиска ‘\_\\_Р ’ буд ет соотв етств ов ать, как и ранее, любому набору симв олов в п ров еряемой строке. О д нако в торой симв ол “**\_**”, след ующ ий п осле симв ола ‘\’, объяв ленного escape-симв олом, уже буд ет интерп ретиров аться букв ально как обыч ный симв ол, так же как и симв ол Р в зад анном шаблоне.

О бращ аем в аше в нимание на то, ч то рассмотренные в ыше оп ераторы срав нения “=, <, >, <=, >=, <>” и оп ераторы **IN**, **BETWEEN** и **LIKE** ни в коем случ ае нельзя исп ользов ать д ля п ров ерки сод ержимого п оля на налич ие в нем п устого знач ения **NULL** (см. разд ел 1.5, Тип ы д анных SQL). Д ля этих целей сп ециально п ред назнач ены оп ераторы **IS** **NULL** (ЯВЛЯЕТСЯ ПУСТЫМ) и **IS** **NOT** **NULL** (ЯВЛЯЕТСЯ НЕ ПУСТЫМ).

**УП РАЖ НЕ НИ Я**

1. Н ап ишите зап рос, в ып олняющ ий в ыв од наход ящ ихся в таблице EXAM\_MARKS номеров п ред метов обуч ения, экзамены п о которым сд ав ались межд у 10 и 20 янв аря 1999 год а.
2. Н ап ишите зап рос, в ыбирающ ий д анные обо в сех п ред метах обуч ения, экзамены п о которым сд аны студ ентами, имеющ ими ид ентификаторы 12 и 32.
3. Н ап ишите з ап рос, который в ып олняет в ыв од назв аний п ред метов обуч ения, нач инающ ихся на букв у ‘И’.
4. Н ап ишите зап рос, в ыбирающ ий св ед ения о студ ентах, у которых имена нач инаются на букв ы ‘И’ или ‘С’.
5. Н ап ишите зап рос д ля в ыбора из таблицы EXAM\_MARKS зап исей, д ля которых отсутств уют знач ения оценок (п оле MARK).
6. Н ап ишите зап рос, в ып олняющ ий в ыв од из таблицы EXAM\_MARKS з ап исей, д ля которых в п оле MARK п ростав лены знач ения оценок.

**2.3. П реобразование вы вода и встроенны е**

**ф ункц ии**

В SQL реализов а ны оп ераторы п реобразов ания д анных и в строенные функции, п ред наз нач енные д ля работы со знач ениями столбцов и/или константами в в ыражениях. Исп ольз ов ание этих оп ераторов д оп устимо в зап росах в езд е, гд е можно исп ольз ов ать в ыражения.

**2.3.1. Числ овы е, символ ь ны е и ст роковы е конст ант ы**

Н есмотря на то, ч то SQL работает с данными в п онятиях строк и столбцов таблиц, имеется в озможность п рименения знач ений в ыражений, п остроенных с исп ользов анием в строенных функций, констант, имен столбцов , которые оп ред еляются как св оего род а в иртуальные столбцы. О ни п омещ аются в сп иске столбцов и могут соп ров ожд аться п сев д онимами.

Если в зап росе в место сп ецификации столбца SQL обнаружив ает *чи сло* , то оно интерп ретируется как *чи сло ва я ко нст а нт а* .

*Си м во льны е ко нст а нт ы* д олжны указыв аться в од инарных кав ыч ка х. Если од инарна я кав ыч ка д олжна в ыв од иться как ч асть строков ой константы, то ее нужно п ред в арить д ругой од инарной кав ыч кой

Н ап ример, результатом в ып олнения зап роса

**SELECT** ‘Фа ми ли я’, SURNAME, ‘Имя’, NAME, 100

**FROM** STUDENT;

яв ляется таблица след ующ его в ид а

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SURNAME |  | NAME |  |
| Фамилия | Ив анов | Имя | Ив ан | 100 |
| Фамилия | Петров | Имя | Петр | 100 |
| Фамилия | Сидоров | Имя | Вадим | 100 |
| Фамилия | К узнецов | Имя | Б орис | 100 |
| Фамилия | Зайцева | Имя | О льга | 100 |
| Фамилия | Пав лов | Имя | А ндрей | 100 |
| Фамилия | К отов | Имя | Пав ел | 100 |
| Фамилия | Л укин | Имя | А ртем | 100 |
| Фамилия | Петров | Имя | А нтон | 100 |
| Фамилия | Б елкин | Имя | Вадим | 100 |
| … … | … … | … … | … … | … … |

**2.3.2. Ариф мет ические операц ии дл я преобраз ования числ овы х данны х**

* У нарный (од иноч ный) оп ератор “–” (з нак минус) изменяет знак ч ислов ого з нач ения, п еред которым он стоит, на п ротив оп оложный.
* Б инарные оп ераторы “**+**”, “**–** “, “**\***” и “**/**” п редостав ляют в оз можность в ып олнения арифметич еских оп ераций сложения, в ыч итания, умножения и д еления.

Н ап ример, результат зап роса

**SELECT** SURNAME, NAME, STIPEND, –(STIPEND\*KURS)/2

**FROM** STUDENT

**WHERE** KURS = 4 **AND** STIPEND > 0; буд ет в ыгляд еть след ующ им образом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SURNAME | NAME | STIPEND | KURS |  |
| Сидоров | Вадим | 150 | 4 | -300 |
| Петров | А нтон | 200 | 4 | -400 |
| … … | … … | … … | … … | … … |

**2.3.3. С им вол ь ная операц ия конкат енац ии ст рок**

О п ерация конкатенации “**||**” п оз в оляет соед инять (“склеив ать”) знач ения д в ух или более столбцов симв ольного тип а или симв ольных констант в од ну

строку.

Эта оп ерация имеет синтаксис

<*зна чи м о е си м во льно е выр а ж ени е*> {**||**}<*зна чи м о е си м во льно е вы р а ж ени е*>.

Н ап ример:

**SELECT** SURNAME **|| ‘\_’ ||** NAME, STIPEND

**FROM** STUDENT

**WHERE** KURS = 4 **AND** STIPEND > 0;

Рез ультат з ап роса буд ет в ыгляд еть след ующ им образом

|  |  |
| --- | --- |
|  | STIPEND |
| Сидоров \_Вадим | 150 |
| Петров \_А нтон | 200 |
| … … | … … |

**2.3.4. С им вол ь ны е ф ункц ии преобраз ования букв раз л ичны х сл ов в ст роке**

* **LOWER** – п ерев од в строч ные симв олы (нижний регистр) **LOWER** (<*ст р о ка* >)
* **UPPER** – п ерев од в п роп исные симв олы (в ерхний регистр) **UPPER** (<*ст р о ка* >)
* **INITCAP** – п ерев од п ерв ой букв ы кажд ого слов а строки в заглав ную

(п роп исную)

**INITCAP**(<*ст р о ка* >)

Н ап ример:

**SELECT** **LOWER**(SURNAME),  **UPPER**(NAME)

**FROM** STUDENT

**WHERE** KURS = 4 **AND** STIPEND > 0;

Рез ультат з ап роса буд ет в ыгляд еть след ующ им образом

|  |  |
| --- | --- |
| SURNAME | NAME |
| Сидоров | ВА Д ИМ |
| Петров | А Н ТО Н |
| … … | … … |

**2.3.5. С им вол ь ны е ст роковы е ф ункц ии**

* **LPAD** – д оп олнение строки слев а

**LPAD** (<*ст р о ка* >, <*д ли на* > [, <*по д ст р о ка* >])

* + <*ст р о ка* > д оп олняется **слев а** указ анной в <*по д ст р о ке*> п ослед ов ательностью симв олов д о указанной <*д ли ны* > (в озможно, с п ов торением п ослед ов ательности);
  + если <*по д ст р о ка* > не указана, то п о умолч анию <*ст р о ка* > д оп олняется п робелами;
  + если <*д ли на* > меньше д лины <*ст р о ки* >, то исход ная <*ст р о ка* > усекается слев а д о зад анной <*д ли ны* >.
* **RPAD** – д оп олнение строки сп рав а

**RPAD** (<*ст р о ка* >, <*д ли на* > [, <*по д ст р о ка* >])

* + <*ст р о ка* > д оп олняется **спра в а** указанной в <*по д ст р о ке*> п ослед ов ательностью симв олов д о указанной <*д ли ны* > (в озможно, с п ов торением п ослед ов ательности);
  + если <*по д ст р о ка* > не указана, то п о умолч анию <*ст р о ка* > д оп олняется п робелами;
  + если <*д ли на* > меньше д лины <*ст р о ки* >, то исход ная <*ст р о ка* > усекается сп рав а д о зад анной <*д ли ны >*.
* **LTRIM** – уд аление лев ых гранич ных симв олов

**LTRIM** (<*ст р о ка* > [,<*по д ст р о ка* >])o из <*ст р о ки* > уд аляются слев а симв олы, указ анные в <*по д ст р о ке*>; o если <*по д ст р о ка* > не указана, то п о умолч анию уд а ляются п робелы; o в <*ст р о ку*> сп рав а д обав ляется столько п робелов , сколько симв олов слев а было уд алено, то есть д лина <*ст р о ки* > остается неизменной.

* **RTRIM** – уд аление п рав ых гранич ных симв олов **RTRIM** (<*ст р о ка* > [,<*по д ст р о ка* >]) o из <*ст р о ки* > уд аляются сп рав а симв олы, указ анные в <*по д ст р оке*>; o если <*по д ст р о ка* > не указана, то п о умолч анию уд аляются п робелы; o в <*ст р о ку*> слев а д обав ляется столько п робелов , сколько симв олов сп рав а было уд алено, то есть д лина <*ст р о ки* > остается неизменной.

Функции **LTRIM** и **RTRIM** рекоменд уется исп ользов ать п ри нап исании услов ных в ыражений, в которых срав нив аются текстов ые строки. Дело в том, ч то налич ие нач альных или конеч ных п робелов в срав нив аемых оп еранд ах может исказ ить результат срав нения.

Н ап ример, константы ‘ ААА’ и ‘ ААА ’ не рав ны д руг д ругу.

* **SUBSTR** – в ыд еление п од строки

**SUBSTR** (<*ст р о ка* >,<*на ча ло* > [,<*ко ли чест во* >])

* + из <*ст р о ки* > в ыбирается зад анное <*ко ли чест во* > симв олов , нач иная с указ анной п озиции в строке <*на ча ло* >;
  + если <*ко ли чест во* > не з ад ано, симв олы в ыбираются с <*на ча ла* > и д о конца <*ст р о ки* >.
  + в озв ращ ается п од строка, сод ержащ ая ч исло симв олов , зад анное п араметром <*ко ли чест во* >, либо ч исло симв олов от п озиции, зад анной п араметром <*на ча ло* > д о конца *ст р о ки* ;
  + если указанное <*на ча ло* > п рев осход ит д лину <*ст р о ки* >, то в озв ращ ается строка, состоящ ая из п робелов . Д лина этой строки буд ет рав на з ад анному <*ко ли чест ву*> или исход ной д лине <*ст р о ки* > (п ри не зад анном <*ко ли чест ве*> ).
* **INSTR** – п оиск п од строки

**INSTR**(<*ст р о ка* >,<*по д ст р о ка* > [,<*на ча ло* *по и ска* >

[,<*но м ер* *вхо ж д ени я*>]])

* + <*на ча ло по и ска* > задает нач альную п оз ицию в строке д ля п оиска

<*по д ст р оки* >. Если не зад ано, то п о умолч анию п ринимается знач ение 1;

* + <*но м ер вхо ж д ени я*> з ад ает п оряд ков ый номер искомой п од строки.

Если не зад ан, то п о умолч анию п р инимается з нач ение 1; o знач имые в ыражения в <*на ча ле пои ска* > или в <*но м ер е вхо ж д ени я*> д олжны иметь безз наков ый целый тип или п рив од иться к этому тип у;

* + тип в озв ращ аемого знач ения – **INT;** o функция в оз в ращ ает п озицию найд енной п од строки.
* **LENGTH –** оп ред еление д лины строки **LENGTH**(<*ст р о ка* >)
  + д лина <*ст р о ки* >, тип в озв ращ аемого знач ения – **INT**; o функция в оз в ращ ает **NULL**, если <*ст р о ка* > имеет **NULL**-знач ение.

**П римеры** зап росов , исп ольз ующ их строков ые функции.

Рез ультат з ап роса

**SELECT** **LPAD** (SURNAME, 10, ‘@’), **RPAD** (NAME, 10, ‘$’)

**FROM** STUDENT

**WHERE** KURS = 3 **AND** STIPEND > 0;

буд ет в ыгляд еть след ующ им образом

|  |  |
| --- | --- |
|  |  |
| @@@@Петров | Петр$$$$$$ |
| @@@@Павлов | Андрей$$$$ |
| @@@@@Лукин | Артем$$$$$ |
| … … | … … |

А зап рос

**SELECT** **SUBSTR**(NAME, 1, 1) **|| ‘.’ ||** SURNAME, CITY, **LENGTH**(CITY)

**FROM** STUDENT

**WHERE** KURS **IN**(2, 3, 4) **AND** STIPEND > 0; в ыд аст результат

|  |  |  |
| --- | --- | --- |
|  | CITY |  |
| П.Петров | К урск | 5 |
| С.Сид оров | Москв а | 6 |
| О .Зайцев а | Л ипецк | 6 |
| А .Лукин | Воронеж | 7 |
| А .Петров | **NULL** | **NULL** |
| … … | … … | … … |

**2.3.6. Ф ункц ии работ ы с числ ам и**

* **ABS** – абсолютное з нач ение

**ABS**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* **FLOOR** – урезает знач ение ч исла с п лав ающ ей точ кой д о наибольшего целого, не п рев осход ящ его зад анное ч исло **FLOOR**(<*зна чи м о е чи сло во е вы р а ж ени е*>)
* **CEIL** – самое малое целое, которое рав но или больше зад анного ч исла

**CEIL**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* Функция округления – **ROUND**

**ROUND**(<*зна чи м о е чи сло во е вы р а ж ени е*>,<*т о чно ст ь*>)аргумент <*т о чно ст ь*> зад ает точ ность округления (см. **пример** ниже)

* Функция усеч ения – **TRUNC**

**TRUNC**(<*зна чи м о е чи сло во е вы р а ж ени е*>,<*т о чно ст ь*>)

* Тригонометрич еские функции – **COS**, **SIN**, **TAN**

**COS**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**SIN**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**TAN**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* Г ип ерболич еские функции – **COSH, SINH, TANH**

**COSH**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**SINH**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**TANH**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* Эксп оненциальная функция – **(EXP)**

**EXP**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* Логарифмич еские функции – **(LN**, **LOG)**

**LN**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**LOG**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

* Функция в озв ед ения в степ ень – **POWER**

**POWER**(<*зна чи м о е чи сло во е вы р а ж ени е*>,<*экспо нент а* >)

* О п ред еление з нака ч исла – **SIGN** **SIGN**(<*зна чи м о е чи сло во е вы р а ж ени е*>)
* Выч исление кв ад ратного корня – **SQRT** **SQRT**(<*зна чи м о е чи сло во е вы р а ж ени е*>)

**П ример**.

Зап рос

**SELECT** UNIV\_NAME, RATING, **ROUND**(RATING, -1), **TRUNC**(RATING, -1) **FROM** UNIVERSITY;

Вернет результат

|  |  |  |  |
| --- | --- | --- | --- |
| UNIV\_NAME | RATING |  |  |
| МГ У | 606 | 610 | 600 |
| ВГ У | 296 | 300 | 290 |
| Н Г У | 345 | 350 | 340 |
| РГ У | 416 | 420 | 410 |
| Б Г У | 326 | 330 | 320 |
| ТГ У | 368 | 370 | 360 |
| ВГ МА | 327 | 330 | 320 |
| … … | … … | … … | … … |

**2.3.7. Ф ункц ии преобраз ования значений**

* Преобраз ов ание в симв ольную строку – **TO\_CHAR**

**TO**\_**CHAR**(<*зна чи м о е вы р а ж ени е*>[,<*си м во льный ф о р м а т* >]) o <*зна чи м о е выр а ж ени е*> д олжно п ред став лять ч ислов ое з нач ение или знач ение тип а д ата-в ремя;

* + д ля ч ислов ых знач ений <*си м во льны й ф о р м а т* > д олжен иметь синтаксис [S]9[9… ][.9[9… ]], гд е S – п ред став ление знака ч исла (п ри отсутств ии п ред п олагается без отображения знака), 9 – п ред став ление цифр-знаков ч ислов ого знач ения (д ля кажд ого знакоместа). Симв ольный формат оп ред еляет в ид отображения ч исел. По умолч анию д ля ч ислов ых знач ений исп ользуется формат

'999999.99'; o д ля знач ений тип а **ДАТ А-В РЕ МЯ** <*си м во льны й ф о р м а т* > имеет в ид (то есть в ид отображения знач ений д аты и в ремени):

* + - в ч асти д аты

‘DD-Mon-YY’

‘DD-Mon-YYYY’

‘MM/DD/YY’

‘MM/DD/YYYY’

‘DD.MM.YY’

‘DD.MM.YYYY’

* + - в ч асти в ремени ‘HH24’ ‘HH24:MI’

‘HH24:MI:SS’

‘HH24:MI:SS.FF’ гд е:

Н Н 24 - ч асы в д иап азоне от 0 д о 24

MI – минуты

SS – секунд ы

FF – тики (сотые д оли секунд ы)

При в ыв од е в ремени в кач еств е раз д елителя п о умолч анию исп ользуется д в оеточ ие (**:**), но п ри жела нии можно исп ольз ов ать любой д ругой симв ол.

Воз в ращ аемое з нач ение – симв ольное п ред став ление <*зна чи м о го вы р а ж ени я*> в соотв етств ии с з ад анным <*си м во льны м ф о р м а т о м* > п реобразов ания.

* Преобраз ов ание из симв ольного знач ения в ч ислов ое – **TO\_NUMBER**

**TO**\_**NUMBER**(<*зна чи м о е си м во льно е вы р а ж ени е*>)

При этом <*зна чи м о е си м во льно е вы р а ж ени е*> д олжно зад ав ать симв ольное знач ение ч ислов ого тип а.

* Преобраз ов ание симв ольной строки в д ату – **TO**\_**DATE**

**TO**\_**DATE**(<*зна чи м о е си м во льно е вы р а ж ени е*> [,<*си м во льны й ф о р м а т* >])o <*зна чи м о е си м вольно е вы р а ж ени е*> д олжно з ад ав ать симв ольное знач ение тип а **ДАТ А-В РЕ МЯ** .

* + <*си м во льны й ф о р м а т* > д олжен оп исыв ать п ред став ление знач ения тип а **ДАТ А**-**В РЕ МЯ** в <*зна чи м о м си м во льно м в ыр а ж ени и* >. Доп устимые форматы (в том ч исле и формат п о умолч анию) п рив ед ены в ыше.

Возв ращ аемое знач ение – <*зна чи м о е си м во льно е вы р а ж ени е*> в о в нутреннем п ред став лении. Тип в озв ращ аемого знач ения – **DATE**. О п ерации над знач ениями тип а **DATE**

Н ад знач ениями тип а **DATE** разрешены след ующ ие оп ерации:

* + бинарная оп ерация сложения; o бинарная оп ерация в ыч итания.

В бинарных оп ерациях од ин из оп еранд ов д олжен иметь знач ение отд ельного элемента д аты: только год , или только месяц, или только д ень.

Н ап ример:

п ри д обав лении к д ате ‘22.05.1998’ п яти лет п олуч ится д ата

‘22.05.2003’;

п ри д обав лении к этой же д ате д ев яти месяцев п олуч ится д ата

‘22.02.1998’; п ри д обав лении 10-ти д ней п олуч им ‘01.06.1998’.

При сложении д в ух п олных д ат, нап ример, ‘22.05.1998’ и ‘01.12.2000’ рез ультат неп ред сказуем.

**П ример**.

Зап рос

**SELECT** SURNAME, NAME, BIRTHDAY,

**TO**\_**CHAR**(BIRTHDAY, **‘**DD-Mon-YYYY**’**),

**TO**\_**CHAR**(BIRTHDAY, **‘**DD.MM.YY**’**) **FROM** STUDENT;

Вернет результат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SURNAME | NAME | BIRTHDAY |  |  |
| Ив анов | Ив ан | 3/12/1982 | 3-дек-1982 | 3.12.82 |
| Петров | Петр | 1/12/1980 | 1-дек-1980 | 1.12.80 |
| Сидоров | Вадим | 7/06/1979 | 7-июн-1979 | 7.06.79 |
| К узнецов | Б орис | 8/12/1981 | 8-дек-1981 | 8.12.81 |
| Зайцев а | О льга | 1/05/1981 | 1-май-1981 | 1.05.81 |
| Пав лов | А нд рей | 5/11/1979 | 5-ноя-1979 | 5.11.79 |
| К отов | Пав ел | **NULL** | **NULL** | **NULL** |
| Л укин | А ртем | 1/12/1981 | 1-дек1981 | 1.12.81 |
| Петров | А нтон | 5/08/1981 | 5-ав г-1981 | 5.08.81 |
| Б елкин | Вадим | 7/01/1980 | 7-янв -1980 | 7.01.80 |
| … … | … … | … … | … … | … … |

Функция **CAST** яв ляется сред ств ом яв ного п реобразов ания д анных из од ного тип а в д ругой. Синтаксис этой команд ы имеет в ид **CAST** <*зна чи м о е вы р а ж ени е*> **AS** <*т и п д а нны х*>

* <*зна чи м о е вы р а ж ени е*> должно иметь ч ислов ой или симв ольный тип языка SQL (в озможно, с указ анием д лины, точ ности и масштаба) или быть **NULL**-знач ением.
* любое ч ислов ое в ыражение может быть яв но п реобразов ано в любой д ругой ч ислов ой тип .
* симв ольное в ыражение может быть п реобразов ано в любой ч ислов ой тип . При этом в рез ультате симв ольного в ыражения отсекаются нач альные и конеч ные п робелы, а остальные симв олы п реобраз уются в ч ислов ое з нач ение п о п рав илам языка SQL.
* если яв но зад анная д лина симв ольного тип а нед остаточ на и п реобразов анное знач ение не размещ ается в нем, то результатив ное з нач ение усекается сп рав а.
* в озможно яв ное п реобразов ание симв ольного тип а в симв ольный с д ругой д линой. Если д лина результата больше д лины аргумента, то знач ение д оп олняется п робелами; если меньше, то усекается.
* **NULL**-знач ение п реобраз уется в **NULL**-з нач ение соотв етств ующ его тип а.
* ч ислов ое в ыражение может быть п реобразов ано в симв ольный тип .

Пример.

**SELECT CAST** STUDENT\_ID **AS** **CHAR**(10)

**FROM** STUDENT;

### УП РАЖ НЕ НИ Я

1. Состав ьте з ап рос д ля таблицы STUDENT таким образ ом, ч тобы в ыход ная таблица сод ержала од ин столбец, сод ержащ ий п ослед ов ательность разд еленных симв олом “;” (точ ка с зап ятой) з нач ений в сех столбцов этой таблицы, и п р и этом текстов ые з нач ения д олжны отображаться

п роп исными симв олами (в ерхний регистр), то есть быть п ред став ленными в след ующ ем в ид е:

10;КУЗНЕЦОВ;БОРИС;0;БРЯНСК;8/12/1981;10.

1. Состав ьте з ап рос д ля таблицы STUDENT таким образом, ч тобы в ыход ная таблица сод ержала в сего од ин столбец в след ующ ем в ид е: Б.КУЗНЕЦОВ; место жительства-БРЯНСК; родился - 8.12.81.
2. Состав ьте з ап рос д ля таблицы STUDENT таким образом, ч тобы в ыход ная таблица сод ержала в сего од ин столбец в след ующ ем в ид е:

б.кузнецов; место жительства-брянск; родился: 8-дек-1981.

1. Состав ьте з ап рос д ля таблицы STUDENT таким образом, ч тобы в ыход ная таблица сод ержала в сего од ин столбец в след ующ ем в ид е:

Борис Кузнецов родился в 1981 году.

1. Выв ести фа милии, имена студ ентов и в елич ину п олуч аемых ими стип енд ий, п ри этом знач ения стип енд ий д олжны быть ув елич ены в 100 раз.
2. То же, ч то и в зад ач е 4, но только д ля студ ентов 1, 2 и 4-го курсов и таким образом, ч тобы фамилии и имена были в ыв ед ены п роп исными букв ами.
3. Состав ьте з ап рос д ля таблицы UNIVERSITY таким образом, ч тобы в ыход ная таблица сод ержала в сего од ин столбец в след ующ ем в ид е: Код-10; ВГУ-г.ВОРОНЕЖ; Рейтинг=296.
4. Тоже, ч то и в зад ач е 7, но знач ения рейтинга требуется округлить д о п ерв ого знака (нап ример, знач ение 382 округляется д о 400).

**2.4. Агрегирование и групповы е ф ункц ии**

А грегирующ ие функции п озв оляют п олуч ать из таблицы св од ную (агрегиров анную) информацию, в ып олняя оп ерации над груп п ой строк таблицы. Для з ад ания в **SELECT-**зап росе агрегирующ их оп ераций исп ользуются след ующ ие ключ ев ые слов а:

* + - **COUNT** оп ред еляет колич еств о строк или знач ений п оля, в ыбранных п осред ств ом з ап роса, и не яв ляющ ихся **NULL**-знач ениями;
    - **SUM** – в ыч исляет арифметич ескую сумму в сех в ыбранных знач ений д анного п оля;
    - **AVG** в ыч исляет сред нее знач ение д ля в сех в ыбранных знач ений д анного п оля;
    - **MAX** в ыч исляет на ибольшее из в сех в ыбранных з нач ений д анного п оля; • **MIN** в ыч исляет на именьшее из в сех в ыбранных знач ений д анного п оля.

В **SELECT-**зап росе агрегирующ ие функции исп ользуются аналогич но именам п олей, п ри этом п ослед ние (имена п олей) исп ользуются в кач еств е аргументов этих функций.

Функция **AVG** п ред назнач ена д ля п од сч ета сред него знач ения п оля на множеств е з ап исей таблицы.

Н ап ример, д ля оп ред еления сред него знач ения п оля MARK (оценки) п о в сем зап исям таблицы EXAM\_MARKS можно исп ользов ать з ап рос с функцией **AVG** след ующ его в ид а:

**SELECT AVG**(MARK)

**FROM** EXAM\_MARKS;

Для п од сч ета общ его колич еств а строк в таблице след ует исп ользов ать функцию **COUNT** со з в ез д оч кой.

**SELECT COUNT**(\*)

**FROM** EXAM\_MARKS;

А ргументы **DISTINCT** и **ALL** п озв оляют, соотв етств енно, исключ ать и в ключ ать д убликаты обрабатыв аемых функцией **COUNT** знач ений, п ри этом необход имо уч итыв ать, ч то п ри исп ольз ов ании оп ции **ALL** знач ения **NULL** в се рав но не в ойд ут в ч исло п од сч итыв аемых знач ений.

**SELECT COUNT**(**DISTINCT** SUBJ\_ID)

**FROM** SUBJECT;

Пред ложение **GROUP BYGROUP BY** (ГРУППИРОВАТЬ ПО) п озв оляет груп п иров ать зап иси в п од множеств а, оп ред еляемые знач ениями какого-либо п оля, и п рименять агрегирующ ие функции уже не ко в сем зап исям таблицы, а разд ельно к кажд ой сформиров анной груп п е.

Пред п оложим, требуется найти максима льное знач ение оценки, п олуч енной кажд ым студ ентом. Зап рос буд ет в ыгляд еть след ующ им образ ом:

**SELECT** STUDENT\_ID, **MAX(**MARK**) FROM** EXAM\_MARKS

**GROUP** **BY** STUDENT\_ID;

Выбираемые из таблицы EXAM\_MARKS зап иси груп п ируются п о знач ениям п оля STUDENT\_ID, указанного в п ред ложении **GROUP BY**, и д ля кажд ой груп п ы наход ится максимальное знач ение п оля MARK. Пред ложение **GROUP BY** п озв оляет п рименять агрегирующ ие функции к кажд ой груп п е, оп ред еляемой общ им з нач ением п оля (или п олей), указанных в этом п ред ложении. В п рив ед енном зап росе рассматрив аются гр уп п ы зап исей, сгруп п иров анные п о ид ентификаторам студ ентов .

В конструкции **GROUP BY** д ля груп п иров ания может быть исп ольз ов ано более од ного столбца. Н ап ример:

**SELECT** STUDENT\_ID, SUBJ\_ID, **MAX(**MARK**)**  **FROM** EXAM\_MARKS

**GROUP** **BY** STUDENT\_ID, SUBJ\_ID;

В этом случ ае строки в нач але груп п ируются п о знач ениям п ерв ого столбца, а в нутри этих груп п – в п од груп п ы п о з нач ениям в торого столбца. Таким образом, **GROUP BY** не только устанав лив ает столбцы, п о которым осущ еств ляется груп п иров ание, но и указыв ает п оряд ок разбиения столбцов на груп п ы.

След ует иметь в в ид у, ч то в п ред ложении **GROUP BY** д олжны быть указаны в се в ыбираемые столбцы, п рив ед енные п осле ключ ев ого слов а **SELECT**, кроме столбцов , указанных в кач еств е аргумента в агрегирующ ей функции.

При необход имости ч асть сформиров анных с п омощ ью **GROUP BY** груп п может быть исключ ена с п омощ ью п ред ложения **HAVING**.

Пред ложение **HAVING** оп ред еляет критерий, п о которому груп п ы след ует в ключ ать в в ыход ные д анные, п о аналогии с п ред ложением **WHERE**, которое осущ еств ляет это д ля отд ельных строк.

**SELECT** SUBJ\_NAME, **MAX**(HOUR)

**FROM** SUBJECT  **GROUP BY** SUBJ\_NAME

**HAVING MAX**(HOUR) >= 72;

В услов ии, з ад ав аемом п ред ложением **HAVING**, указыв ают только п оля или в ыражения, которые на в ыход е имеют ед инств енное знач ение д ля кажд ой в ыв од имой груп п ы.

### УП РАЖ НЕ НИ Я

1. Н ап ишите зап рос д ля п од сч ета колич еств а студ ентов , сд ав ав ших экзамен п о п ред мету обуч ения с ид ентификатором, рав ным 20.
2. Н ап ишите зап рос, который п озв оляет п од сч итать в таблице EXAM\_MARKS колич еств о различ ных п ред метов обуч ения.
3. Н ап ишите зап рос, который в ып олняет в ыборку д ля кажд ого студ ента

з нач ения его ид ентификатора и минима льной из п олуч енных им оценок.

1. Н ап ишите зап рос, который в ып олняет в ыборку д ля кажд ого студ ента з нач ения его ид ентификатора и максимальной из п олуч енных им оценок.
2. Н ап ишите зап рос, в ып олняющ ий в ыв од фамилии п ерв ого в алфа в итном п оряд ке (п о фамилии) студ ента, фамилия которого нач инается на букв у “И”.
3. Н ап ишите зап рос, который в ып олняет в ыв од д ля кажд ого п ред мета обуч ения наименов ание п ред мета и максимальное знач ение номера семестра, в котором этот п ред мет п реп од ается.
4. Н ап ишите з ап рос, который в ып олняет в ыв од д анных д ля кажд ого конкретного д ня сд ач и экзамена о колич еств е студ ентов , сд ав ав ших экзамен в этот д ень.
5. Н ап ишите зап рос д ля п олуч ения сред него балла д ля кажд ого курса п о кажд ому п ред мету.
6. Н ап ишите з ап рос д ля п олуч ения сред него балла д ля кажд ого студ ента.
7. Н ап ишите з ап рос д ля п олуч ения сред него балла д ля кажд ого экзамена.
8. Н ап ишите зап рос д ля оп ред еления колич еств а студ ентов , сд ав ав ших кажд ый экзамен.
9. Н ап ишите зап рос д ля оп ред еления колич еств а изуч аемых п ред метов на кажд ом курсе.

**2.5. П уст ы е значения (NULL) в агрегирую щ их ф ункц ия х**

Н алич ие п устых (**NULL**) знач ений в п олях таблицы наклад ыв ает особенности на в ып олнение агрегирующ их оп ераций над д анными, которые след ует уч итыв ать п ри их исп ользов а нии в SQL-зап росах.

### 2.5.1. В л ия ние NULL– з начений в ф ункц ии COUNT

Если аргумент функции **COUNT** яв ляется константой или столбцом без п устых з нач ений, то функция в озв ращ ает колич еств о строк, к которым п рименимо оп ред еленное услов ие или груп п иров ание.

Если аргументом функции яв ляется *ст о лбец*, сод ержащ ий п устое знач ение, то **COUNT** в ернет ч исло строк, не сод ержащ их п устые з нач ения, и к которым п рименимо оп ред еленное услов ие или груп п иров ание.

Если бы механиз м **NULL** не был д оступ ен, то неп рименимые и отсутств ующ ие з нач ения п ришлось бы исключ ать с п омощ ью конструкции **WHERE**.

Пов ед ение функции **COUNT**(\*) не з ав исит от п устых з нач ений. О на в озв ратит общ ее колич еств о строк в таблице.

### 2.5.2. В л ия ние NULL-з начений в ф ункц ии AVG

*Ср ед нее* знач ение множеств а ч исел рав но сумме ч исел, д еленной на ч исло элементов множеств а. О д нако, если некоторые элементы п усты, то есть их знач ения неизв естны или не сущ еств уют, то д еление на колич еств о в сех элементов множеств а п рив ед ет к неп рав ильному рез ультату.

Функция **AVG** в ыч исляет сред нее знач ение в сех *и звест ны х* знач ений множеств а элементов , то есть эта функция п од сч итыв ает сумму *и звест ны х* знач ений и д елит ее на колич еств о *эт и х* з нач ений, а не на общ ее колич еств о знач ений, сред и которых могут быть **NULL**-знач ения. Если столбец состоит только из п устых з нач ений, то функция **AVG** также в озв ратит **NULL**.

**2.6. Рез ул ь т ат дей ствия трехзначны х усл овны х операторов**

У слов ные оп ераторы п ри отсутств ии п устых знач ений в озв ращ ают либо **TRUE** (и сти н а ), либо **FALSE** (ло ж ь). Если же в столбце п рисутств уют п устые знач ения, то может быть в озв ращ ено и третье з нач ение: **UNKNOWN** (н еи звестн о ). В этой схеме, нап ример, услов ие **WHERE** A=2, гд е А – имя столбца, знач ения которого могут быть неиз в естны, п ри A=2 буд ет соотв етств ов ать **TRUE**, п ри A=4 в результате буд ет п олуч ено знач ение **FALSE**, а п ри отсутств ующ ем знач ении A (**NULL**-з нач ение) результат буд ет **UNKNOWN**. Пустые знач ения оказ ыв ают в лияние на исп ользов ание логич еских оп ераторов **NOT**, **AND** и **OR**.

**Опера тор NOT**

О быч ный унарный оп ератор **NOT** обращ ает оценку **TRUE** в **FALSE** и наоборот. О д нако **NOT** **NULL** п о п режнему буд ет в озв ращ ать п устое знач ение **NULL**. При этом след ует отлич ать случ ай **NOT** **NULL** от услов ия **IS** **NOT** **NULL**, которое яв ляется п ротив оп оложностью **IS** **NULL**, отд еляя изв естные знач ения от неиз в естных.

**Опера тор AND**

* Если результат д в ух услов ий, объед иненных оп ератором **AND**, из в естен, то п рименяются п рав ила булев ой логики, то есть п ри обоих утв ержд ениях **TRUE** состав ное утв ержд ение также буд ет **TRUE**. Если же хотя бы од но из д в ух утв ержд ений буд ет **FALSE**, то состав ное утв ержд ение буд ет **FALSE**.
* Если результат од ного из утв ержд ений неиз в естен, а д ругой оценив ается как **TRUE**, то состояние неизв естного утв ержд ения яв ляется оп ред еляющ им, и, след ов ательно, итогов ый рез ультат также неизв естен.
* Если результат од ного из утв ержд ений неиз в естен, а д ругой оценив ается как **FALSE**, итогов ый результат буд ет **FALSE**.
* Если результат обоих утв ержд ений неиз в естен, то рез ультат также остается неизв естным.

**Опера тор OR**

* Если результат д в ух услов ий, объед иненных оп ератором **OR**, из в естен, то п рименяются п рав ила булев ой логики, а именно: если хотя бы од но из д в ух утв ержд ений соотв етств ует **TRUE**, то и состав ное утв ержд ение буд ет **TRUE**, если оба утв ержд ения оценив а ются как **FALSE**, то состав ное утв ержд ение буд ет **FALSE**.
* Если результат од ного из утв ержд ений неиз в естен, а д ругой оценив ается как **TRUE**, итогов ый результат буд ет **TRUE**.
* Если результат од ного из утв ержд ений неиз в естен, а д ругой оценив ается как **FALSE**, то состояние неиз в естного утв ержд ения играет роль. След ов ательно, итогов ый результат также неизв естен.
* Если результат обоих утв ержд ений неиз в естен, то рез ультат также остается неизв естным.

**П рим ечание.**

О тсутств ующ ие (**NULL**) знач ения целесообразно исп ользов ать в столбцах, п ред назнач енных д ля агрегиров ания, ч тобы из в леч ь п реимущ еств а из сп особа обработки п устых з нач ений в функциях **COUNT** и **AVG**. Практич ески в о в сех остальных случ аях п устых знач ений след ует избегать, так как п ри их налич ии сущ еств енно усложняется корректное п остроение услов ий отбора, п рив од я иногд а к неп ред сказ уемым рез ультатам в ыборки. Для инд икации же отсутств ующ их, неп рименимых или п о какой-то п рич ине неизв естных д анных можно исп ользов ать знач ения п о умолч анию, устанав лив аемые заранее (нап ример, с п омощ ью команд ы **CREATE** **TABLE**

(раз д ел 4.1).

**2.7. Упоря дочение вы ходны х пол ей (ORDER BY)**

К ак уже отмеч алось, з ап иси в таблицах реляционной базы д анных неуп оряд оч ены. О д нако, д анные, в ыв од имые в результате в ып олнения зап роса, могут быть уп оряд оч ены. Для этого исп ольз уется оп ератор **ORDER** **BY**, который п озв оляет уп оряд оч ив ать в ыв од имые зап иси в соотв етств ии со знач ениями од ного или нескольких в ыбранных столбцов . При этом можно зад ать в оз растающ ую (**ASC**) или убыв ающ ую (**DESC**) п ослед ов ательность сортиров ки д ля кажд ого из столбцов . По умолч анию п ринята в озрастающ ая п ослед ов ательность сортиров ки.

Зап рос, п оз в оляющ ий в ыбрать в се д анные из таблицы п ред метов обуч ения SUBJECT, с уп оряд оч ив анием п о наименов аниям п ред метов , в ыгляд ит след ующ им образом:

**SELECT** \*

**FROM** SUBJECT

**ORDER BY** SUBJ\_NAME;

Тот же сп исок, но уп оряд оч енный в обратном п оряд ке, можно п олуч ить зап росом:

**SELECT** \*

**FROM** SUBJECT

**ORDER BY** SUBJ\_NAME **DESC**;

М ожно уп оряд оч ить в ыв од имый сп исок п ред метов обуч ения п о знач ениям семестров , а в нутри семестров – п о наименов а ниям п ред метов .

**SELECT** \*

**FROM** SUBJECT

**ORDER BY** SEMESTR, SUBJ\_NAME;

Пред ложение **ORDER BY** может исп ользов аться с **GROUP** **BY** д ля уп оряд оч ив ания груп п зап исей. При этом оп ератор **ORDER BY** в зап росе *всегд а д о лж ен бы т ь* ***последним***.

**SELECT** SUBJ\_NAME, SEMESTR, **MAX**(HOUR)

**FROM** SUBJECT

**GROUP BY** SEMESTR, SUBJ\_NAME

**ORDER BY** SEMESTR;

При уп оряд оч ив ании в место наименов а ний столбцов можно указ ыв ать их номера, имея, од нако, в в ид у, ч то в д анном случ ае это – номера столбцов , указ анные п ри оп ред елении в ыход ных д анных в зап росе, а не номера столбцов в таблице. Полем с номером 1 яв ляется п ерв ое п оле, указанное в п ред ложении **ORDER** **BY** – незав исимо от его расп оложения в таблице.

**SELECT** SUBJ\_ID, SEMESTR  **FROM** SUBJECT

**ORDER BY** 2 **DESC**;

В этом зап росе в ыв од имые зап иси буд ут уп оряд оч ены п о п олю SEMESTR.

Если в п оле, которое исп ользуется д ля уп оряд оч ив ания, сущ еств уют **NULL-**з нач ения, то в се они размещ аются в конце или п ред шеств уют в сем остальным з нач ениям этого п оля.

### УП РАЖ НЕ НИ Я

1. Пред п оложим, ч то стип енд ия в сем студ ентам ув елич ена на 20%. Н ап ишите зап рос к таблице STUDENT, в ып олняющ ий в ыв од номера студ ента, фамилию студ ента и в елич ину ув елич енной стип енд ии. Выход ные д анные уп оряд оч ить: а) п о знач ению п ослед него столбца (в елич ине стип енд ии); б) в алфав итном п оряд ке фамилий студ ентов .
2. Н ап ишите зап рос, который п о таблице EXAM\_MARKS п озв оляет найти а) максимальные и б) минимальные оценки кажд ого студ ента и в ыв од ит их в месте с ид ентификатором студ ента.
3. Н ап ишите зап рос, в ып олняющ ий в ыв од сп иска п ред метов обуч ения в п оряд ке а) убыв ания семестров и б) в озрастания отв од имых на п ред мет ч асов . Поле семестра в в ыход ных д а нных д олжно быть п ерв ым, з а ним д олжны след ов ать имя п ред мета обуч ения и ид ентификатор п ред мета.
4. Н ап ишите зап рос, который в ып олняет в ыв од суммы баллов в сех студ ентов д ля кажд ой д аты сд ач и экз аменов и п ред став ляет рез ультаты в п оряд ке убыв ания этих сумм.
5. Н ап ишите зап рос, который в ып олняет в ыв од а) сред него, б) минимального, в ) максимального баллов в сех студ ентов д ля кажд ой д аты сд ач и экзаменов , и п ред став ляет результаты в п оряд ке убыв ания этих з нач ений.

**2.8. В л ож енны е подзапросы**

SQL п оз в оляет исп ользов ать од ни зап росы в нутри д ругих зап росов , то есть в клад ыв ать зап росы д руг в д руга. Пред п оложим, из в естна фамилия студ ента (“Петр о в”), но неиз в естно знач ение п оля STUDENT\_ID д ля него. Чтобы изв леч ь д анные обо в сех оценках этого студ ента, можно зап исать след ующ ий зап рос:

**SELECT** \*

**FROM** EXAM\_MARKS

**WHERE** STUDENT\_ID =

( **SELECT** STUDENT\_ID

**FROM** STUDENT SURNAME = ‘Петр о в’);

К ак работает зап рос SQL со св яз анным п од зап росом?

* Выбирается строка из таблицы, имя которой указано в о в нешнем зап росе.
* Вып олняется п од зап рос и п олуч енное в рез ультате его в ып олнения знач ение п рименяется д ля анализ а этой строки в услов ии п ред ложения **WHERE** в нешнего з ап роса.
* По результату оценки этого услов ия п ринимается решение о в ключ ении или не в ключ ении строки в состав в ыход ных д анных.
* Процед ура п ов торяется д ля след ующ ей строки таблицы в нешнего зап роса.

След ует обратить в нимание, ч то п рив ед енный в ыше зап рос корректен только в том случ ае, если в результате в ып олнения указ анного в скобках *по д* зап роса в озв ращ ается *ед и нст венно е зна чени е*. Если в результате в ып олнения п од зап роса буд ет в озв ращ ено несколько знач ений, то этот п од зап рос буд ет ошибоч ным. В д анном п римере это п роизойд ет, если в таблице STUDENT буд ет несколько зап исей со знач ениями п оля SURNAME = ‘Петр о в’.

В некоторых случ аях д ля гарантии п олуч ения ед инств енного з нач ения в результате в ып олнения п од зап роса исп ольз уется **DISTINCT**. О д ним из в ид ов функций, которые ав томатич ески *всегд а* в ыд ают в рез ультате ед инств енное знач ение д ля любого колич еств а строк, яв ляются агрегирующ ие функции.

О п ератор **IN** также широко п рименяется в п од з ап росах. О н зад ает сп исок знач ений, с которыми срав нив аются д ругие знач ения д ля оп ред еления истинности зад ав аемого этим оп ератором п ред иката.

Данные обо в сех оценках (таблица EXAM\_MARKS) студ ентов из Воронежа можно в ыбрать с п омощ ью след ующ его зап роса:

**SELECT** \*

**FROM** EXAM\_MARKS

**WHERE** STUDENT\_ID **IN**

( **SELECT** STUDENT\_ID

**FROM** STUDENT

**WHERE** CITY = ‘Во р о н еж ’);

Под зап росы можно п рименять в нутри п ред ложения **HAVING**. Пусть требуется оп ред елить колич еств о п ред метов обуч ения с оценкой, п рев ышающ ей сред нее знач ение оценки студ ента с ид ентификатором 301:

**SELECT** **COUNT**(**DISTINCT** SUBJ\_ID), MARK

**FROM** EXAM\_MARKS

**GROUP BY** MARK

**HAVING** MARK >

( **SELECT AVG**(MARK)

**FROM** EXAM\_MARKS

**WHERE** STUDENT\_ID = 301);

**2.9. Ф орм ирование свя занны х подзапросов**

При исп ольз ов ании п од зап росов в о в нутреннем зап росе можно ссылаться на таблицу, имя которой указ ано в п ред ложении **FROM** в нешнего зап роса. В этом случ ае такой *связа нны й* п од зап рос в ып олняется п о од ному разу д ля *ка ж д о й* строки таблицы основ ного зап роса.

**П ример**: в ыбрать св ед ения обо в сех п ред метах обуч ения, п о которым п ров од ился экзамен 20 янв аря 1999 г.

**SELECT** \*

**FROM** SUBJECT SU

**WHERE** ‘20/01/1999’ **IN**

( **SELECT** EXAM\_DATE

**FROM** EXAM\_MARKS EX

**WHERE** SU.SUBJ\_ID = EX.SUBJ\_ID);

В некоторых СУ Б Д д ля в ып олнения этого зап роса, в оз можно, п отребуется п реобразов ание знач ения д аты в симв ольный тип . В п рив ед енном зап росе SU и EX яв ляются п сев д онимами (алиасами), то есть сп ециально в в од имыми именами, которые могут быть исп ользов аны в д анном зап росе в место настоящ их имен. В п рив ед енном п римере они исп ользуются в место имен таблиц SUBJECT и EXAM\_MARKS. .

Эту же зад ач у можно решить с п омощ ью оп ерации соед инения таблиц:

**SELECT** **DISTINCT** SU.SUBJ\_ID, SUBJ\_NAME, HOUR, SEMESTER

**FROM** SUBJECT FIRST, EXAM\_MARKS SECOND

**WHERE** FIRST.SUBJ\_ID = SECOND.SUBJ\_ID  **AND** SECOND.EXAM\_DATE = ’20/01/1999’;

В этом в ыражении алиасами таблиц яв ляются имена FIRST и SECOND.

Можно исп ользов ать п од зап росы, св языв ающ ие таблицу со св оей собств енной коп ией. Н а п ример, над о на йти ид ентификаторы, фамилии и стип енд ии студ ентов , п олуч ающ их стип енд ию в ыше сред ней на курсе, на котором они уч атся.

**SELECT** **DISTINCT** STUDENT\_ID, SURNAME, STIPEND  **FROM** STUDENT E1

**WHERE** STIPEND >

**(SELECT AVG**(STIPEND)

**FROM** STUDENT E2

**WHERE** E1.KURS = E2.KURS);

Тот же результат можно п олуч ить с п омощ ью след ующ его зап роса:

**SELECT DISTINCT** STUDENT\_ID, SURNAME, STIPEND

**FROM** STUDENT E1,

**(SELECT** KURS, **AVG**(STIPEND) **AS** AVG\_STIPEND

**FROM** STUDENT E2

**GROUP BY** E2.KURS)E3

**WHERE** E1.STIPEND > AVG\_STIPEND **AND** E1.KURS=E3.KURS;

О братите в нимание – в торой зап рос буд ет в ып олнен горазд о быстрее. Дело в том, ч то в п ерв ом в ариа нте зап роса агрегирующ ая функция **AVG** в ып олняется над таблицей, указанной в п од зап росе, д ля *ка ж д о й* строки в нешнего зап роса. В д ругом в арианте в торая таблица (алиас E2) обрабатыв ается агрегирующ ей функцией од ин раз, в рез ультате ч его формируется в сп омогательная таблица (в зап росе она имеет алиас E3), со строками которой затем соед иняются строки п ерв ой таблицы (алиас E1). След ует иметь в в ид у, ч то реальное в ремя в ып олнения зап роса в большой степ ени з ав исит от оп тимизатора з ап росов конкретной СУ Б Д.

**2.10. С вя занны е подзапросы в HAVING**

В разд еле 2.4 указ ыв алось, ч то п ред ложение **GROUP** **BY** п озв оляет груп п иров ать в ыв од имые **SELECT**-зап росом з ап иси п о з нач ению некоторого п оля. Исп ольз ов ание п ред ложения **HAVING** п озв оляет п ри в ыв од е осущ еств лять фильтрацию таких груп п . Пред икат п ред ложения **HAVING** оценив ается не д ля кажд ой строки результата, а д ля кажд ой груп п ы в ыход ных зап исей, сформиров анной п ред ложением **GROUP** **BY** в нешнего зап роса.

Пусть, нап ример, необход имо п о д анным из таблицы EXAM\_MARKS оп ред елить сумму п олуч енных студ ентами оценок (знач ений п оля MARK), сгруп п иров ав знач ения оценок п о д атам экзаменов и исключ ив те д ни, когд а ч исло студ ентов , сд ав ав ших в теч ение д ня экзамены, было меньше 10.

**SELECT** EXAM\_DATE, **SUM**(MARK)  **FROM** EXAM\_MARKS A

**GROUP** **BY** EXAM\_DATE

**HAVING** 10 <

( **SELECT** **COUNT**(MARK)

**FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE);

Под зап рос в ыч исляет колич еств о строк с од ной и той же д атой, сов п ад ающ ей с д атой, д ля которой сформиров ана оч еред ная груп п а основ ного зап роса.

### УП РАЖ НЕ НИ Я

1. Н ап ишите зап рос с п од зап росом д ля п олуч ения д анных обо в сех оценках студ ента с фамилией “Ива н о в”. Пред п оложим, ч то его п ерсональный номер не из в естен. Всегд а ли такой зап рос буд ет корректным?
2. Н ап ишите з ап рос, в ыбирающ ий д анные об именах в сех студ ентов , имеющ их п о п ред мету c ид ентификатором 101 балл в ыше общ его сред него балла.
3. Н ап ишите зап рос, который в ып олняет в ыборку имен в сех студ ентов , имеющ их п о п ред мету c ид ентификатором 102 балл ниже общ его сред него балла.
4. Н ап ишите з ап рос, в ып олняющ ий в ыв од колич еств а п ред метов , п о которым экз аменов ался кажд ый студ ент, сд а в ав ший более 20-ти п ред метов .
5. Н ап ишите команд у **SELECT**, исп ольз ующ ую св яз анные п од з ап росы и в ып олняющ ую в ыв од имен и ид ентификаторов студ ентов , у которых стип енд ия сов п ад ает с максимальным знач ением стип енд ии д ля город а, в котором жив ет студ ент.
6. Н ап ишите зап рос, который п озв оляет в ыв ести имена и ид ентификаторы в сех студ ентов , д ля которых точ но изв естно, ч то они п рожив ают в город е, гд е нет ни од ного унив ерситета.
7. Н ап ишите д в а зап роса, которые п озв оляют в ыв ести имена и ид ентификаторы в сех студ ентов , д ля которых точ но из в естно, ч то они п рожив ают не в том город е, гд е расп оложен их унив ерситет. О д ин зап рос с исп ользов анием соед инения, а д ругой – с исп ользов анием св язанного

п од зап роса.

**2.11. И спол ь зование оператора EXISTS**

Исп ользуемый в SQL оп ератор **EXISTS** (СУЩЕСТВУЕТ) генерирует знач ение и сти н а или ло ж ь, п од обно булев у в ыражению. Исп ользуя п од зап росы в кач еств е аргумента, этот оп ератор оценив ает результат в ып олнения п од зап роса как истинный, если этот п од зап рос генерирует в ыход ные д анные, то есть в случ ае *сущест во ва ни я*  (в озв рата) хотя бы од ного найд енного з нач ения. В п ротив ном случ ае результат п од зап роса – ложный. О п ератор **EXISTS** не может п ринимать з нач ение unknown (неизвестно).

Пусть, нап ример, нужно из в леч ь из таблицы EXAM\_MARKS д анные о студ ентах, п олуч ив ших хотя бы од ну неуд ов летв орительную оценку.

**SELECT DISTINCT** STUDENT\_ID

**FROM** EXAM\_MARKS A

**WHERE EXISTS**

( **SELECT** \*

**FROM** EXAM\_MARKS B

**WHERE** MARK < 3

**AND** B.STUDENT\_ID=A.STUDENT\_ID);

При исп ольз ов ании св язанных п од зап росов п ред ложение **EXISTS** анализ ирует кажд ую строку таблицы, на которую имеется ссылка в о в нешнем з ап росе. Г лав ный зап рос п олуч ает строки-канд ид аты на п ров ерку услов ия. Для кажд ой строки-канд ид ата в ып олняется п од зап рос. К ак только п од зап рос наход ит строку, гд е в столбце MARK знач ение уд ов летв оряет услов ию, он п рекращ ает в ып олнение и в оз в ращ ает знач ение и сти н а в нешнему зап росу, который з атем анализ ирует св ою строку-канд ид ата.

Н ап ример, требуется п олуч ить ид ентификаторы п ред метов обуч ения, экзамены п о которым сд ав ались не од ним, а несколькими студ ентами:

**SELECT DISTINCT** SUBJ\_ID  **FROM** EXAM\_MARKS A

**WHERE EXISTS**

( **SELECT** \*

**FROM** EXAM\_MARKS B

**WHERE** A.SUBJ\_ID = B.SUBJ\_ID

**AND** A.STUDENT\_ID < > B.STUDENT\_ID);

Часто **EXISTS** п рименяется с оп ератором **NOT** (п о-русски **NOT** **EXISTS** интерп ретируется, как “*не сущест вует …* ”). Если п ред ыд ущ ий з ап рос сформулиров ать след ующ им образ ом – найти ид ентификаторы п ред метов обуч ения, которые сд ав ались од ним и только од ним студ ентом (д ругими слов ами, д ля которых не сущ еств ует д ругого сд ав ав шего студ ента), то д остаточ но п росто п остав ить **NOT** п еред **EXISTS**.

След ует иметь в в ид у, ч то в п од зап росе, указыв аемом в оп ераторе **EXISTS**, *нельзя и спо льзо ва т ь а гр еги р ующи е ф ункци и* .

Возможности п рименения в ложенных зап росов в есьма раз нообраз ны. Н ап ример, п усть из таблицы STUDENT требуется из в леч ь строки д ля кажд ого студ ента, сд ав шего более од ного п ред мета.

**SELECT** \*

**FROM** STUDENT FIRST

**WHERE EXISTS**

(**SELECT** SUBJ\_ID

**FROM** EXAM\_MARKS SECOND **GROUP** **BY** SUBJ\_ID

**HAVING** **COUNT**(SUBJ\_ID) >1

**WHERE** FIRST.STUDENT\_ID = SECOND.STUDENT\_ID);

### УП РАЖ НЕ НИ Я

1. Н ап ишите зап рос с **EXISTS**, п оз в оляющ ий в ыв ести д анные обо в сех студ ентах обуч ающ ихся в в узах, имеющ их рейтинг в ыше 300.
2. Н ап ишите п ред ыд ущ ий з ап рос, исп ользуя соед инения.
3. Н ап ишите за п рос с **EXISTS**, в ыбирающ ий св ед ения обо в сех студ ентах, д ля которых в том же город е, гд е жив ет студ ент, сущ еств уют унив ерситеты, в которых он не уч ится.
4. Н ап ишите з ап рос, в ыбирающ ий из таблицы SUBJECT д анные о назв аниях п ред метов обуч ения, экз амены п о которым *сд а ны* более ч ем од ним студ ентом.

**2.12. О ператоры сравнения с множ еством значений IN, ANY, ALL**

О п ераторы срав нения с множеств ом з нач ений имеют след ующ ий смысл.

|  |  |
| --- | --- |
| **IN** | *Ра вно* любому из знач ений, п олуч енных в о в нутреннем з ап росе. |
| **NOT IN** | *Не р а вно* ни од ному из знач ений, п олуч енных в о в нутреннем зап росе. |
| **=** **ANY** | То же, ч то и **IN**. Соотв етств ует логич ескому оп ератору **OR**. |
| **>** **ANY**, **>** **=** **ANY** | *Бо льш е, чем* (либо *бо льш е и ли р а вно* ) любое п олуч енное ч исло. Экв ив алентно > или > = д ля самого меньшего п олуч енного ч исла. |
| **<** **ANY**, **<** **=** **ANY** | *М еньш е, чем* (либо *м еньш е и ли р а вно* ) любое п олуч енное ч исло. Экв ив алент < или < = д ля самого большего п олуч енного ч исла. |
| **=** **ALL** | Рав но в сем п олуч енным знач ениям. Экв ив алентно логич ескому оп ератору **AND**. |
| **>** **ALL**, **>** **=** **ALL** | *Бо льш е, чем* (либо *бо льш е и ли р а вно* ) в се п олуч енные ч исла. Экв ив алент > или > = д ля самого большего п олуч енного ч исла. |
| **<** **ALL**, **<** **=** **ALL** | *М еньш е, чем* (либо *м еньш е и ли р а вно* ) в се п олуч енные ч исла. Экв ив алентно < или < = самого меньшего п олуч енного ч исла. |

След ует иметь в в ид у, ч то в некоторых СУ Б Д п од д ержив аются не в се из этих оп ераторов .

**П рим еры запросов с испол ь зованием приведенны х операторов.**

Выбрать св ед ения о студ ентах, п рожив а ющ их в город е, гд е расп оложен унив ерситет, в котором они уч атся.

**SELECT** \*

**FROM** STUDENT S

**WHERE** CITY = **ANY**

( **SELECT** CITY

**FROM** UNIVERSITY U

**WHERE** U.UNIV\_ID = S.UNIV\_ID);

Другой в ариант этого з ап роса

**SELECT** \*

**FROM** STUDENT S

**WHERE** CITY **IN**

(**SELECT** CITY

**FROM** UNIVERSITY U

**WHERE** U.UNIV\_ID = S.UNIV\_ID);

Выборка д анных об ид ентификаторах студ ентов , у которых оценки п рев осход ят в елич ину, п о крайней мере, од ной из оценок, п олуч енных ими же 6 октября 1999 год а.

**SELECT** **DISTINCT** STUDENT\_ID

**FROM** EXAM\_MARKS

**WHERE** MARK > **ANY** (**SELECT** MARK

**FROM** EXAM\_MARKS

**WHERE** EXAM\_DATE = ‘06/10/1999’);

О п ератор **ALL**, как п рав ило, эффектив но исп ольз уется с нерав енств ами, а не с рав енств ами, п оскольку знач ение *р а вно* *всем* , которое должно п олуч иться в этом случ ае в результате в ып олнения п од зап роса, может иметь место, только если в се результаты ид ентич ны. Такая ситуация п рактич ески не может быть реализ ов ана, так как, если п од зап рос генерирует множеств о различ ных знач ений, то никакое од но знач ение не может быть рав но сразу в сем з нач ениям в обыч ном смысле. В SQL в ыражение < > **ALL** реально означ ает *не р а вно ни о д но м у* из рез ультатов п од зап роса.

Под зап рос, в ыбирающ ий д анные о наз в аниях в сех унив ерситетов с рейтингом более в ысоким, ч ем рейтинг любого унив ерситета в Воронеже:

**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING > **ALL**

( **SELECT** RATING

**FROM** UNIVERSITY

**WHERE** CITY = ‘Во р о н еж ’);

В этом зап росе в место **ALL** можно также исп ольз ов ать **ANY**. (Проанализируйте, как в этом случ ае изменится смысл п рив ед енного зап роса?)

**SELECT** \*

**FROM** UNIVERSITY

**WHERE NOT** RATING > **ANY**

( **SELECT** RATING

**FROM** UNIVERSITY

**WHERE** CITY = ‘Во р о н еж ’);

**2.13. О собенности прим енения операторов ANY, ALL, EXISTS при обработке пуст ы х значений (NULL)**

Н еобход имо иметь в в ид у, ч то п ри обработке **NULL-**знач ений след ует уч итыв ать различ ие реакции на них оп ераторов **EXISTS**, **ANY** и **ALL**.

К огд а п рав ильный п од зап рос не генерирует никаких в ыход ных д а нных, оп ератор **ALL** ав томатич ески п ринимает з нач ение и сти н а , а оп ератор **ANY** – знач ение ло ж ь.

Зап рос

**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING > **ANY**

( **SELECT** RATING

**FROM** UNIVERSITY

**WHERE** CITY = ‘New York’);

не генерирует в ыход ных д анных (п од раз умев ается, ч то в базе нет д анных об унив ерситетах из город а New York), в то в ремя как зап рос

**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING > **ALL**

( **SELECT** RATING

**FROM** UNIVERSITY

**WHERE** CITY = ‘New York’);

п олностью в осп роизв ед ет таблицу UNIVERSITY.

Исп ользов ание **NULL**-знач ений соз д ает оп ред еленные п роблемы д ля рассматрив аемых оп ераторов . К огд а в SQL срав нив аются д в а знач ения, од но из которых **NULL**-знач ение, результат п ринимает знач ение **UNKNOWN** (неизв естно). Пред икат **UNKNOWN**, также как и **FALSE**-п ред икат, созд ает ситуацию, когд а строка не в ключ ается в состав в ыход ных д анных, но результат п ри этом буд ет различ ен д ля раз ных тип ов зап росов , в зав исимости от исп ольз ов ания в них **ALL** или **ANY** в место **EXISTS**. Рассмотрим в кач еств е п римера д в е реализ ации зап роса: найти в се д анные об унив ерситетах, рейтинг которых меньше рейтинга любого унив ерситета в Москв е.

1. **SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING < **ANY**

(**SELECT** RATING

**FROM** UNIVERSITY

**WHERE** CITY = ‘М о сква ’);

1. **SELECT** \*

**FROM** UNIVERSITY A

**WHERE** **NOT** **EXISTS**

(**SELECT** \*

**FROM** UNIVERSITY B

**WHERE** A.RATING >= B.RATING

**AND** B.CITY = ‘М о сква ’);

При отсутств ии в таблицах **NULL** оба эти зап роса в ед ут себя сов ершенно од инаков о. Пусть теп ерь в таблице UNIVERSITY есть строка с **NULL**знач ениями в столбце RATING. В в ерсии зап роса c **ANY** в основ ном з ап росе, когд а в ыбирается п оле RATING с **NULL**, п ред икат п ринимает знач ение **UNKNOWN** и строка не в ключ ается в состав в ыход ных д а нных. Во в тором же в арианте зап роса, когд а **NOT** **EXISTS** в ыбираетэтустроку в основ номзап росе, **NULL**-знач ение исп ольз уется в п ред икате п од зап роса, п рисв а ив ая ему знач ение **UNKNOWN**. Поэтому в рез ультате в ып олнения п од зап роса не буд ет п олуч ено ни од ного з нач ения и п од зап рос п римет з нач ение *ло ж ь*. Это в св ою оч еред ь сд елает **NOT** **EXISTS** истинным, и, след ов ательно, строка с **NULL** знач ением в п оле RATING п оп ад ет в в ыход ные д анные. По смыслу зап роса такой результат яв ляется неп рав ильным, так как на самом д еле рейтинг унив ерситета, оп исыв аемого д анной строкой может быть и больше рейтинга какого-либо москов ского унив ерситета (он п росто неизв естен). У казанная п роблема св язана с тем, ч то знач ение **EXISTS** в сегд а п ринимает знач ения *и ст и на* или *ло ж ь*, и никогд а – **UNKNOWN**. Это яв ляется д ов од ом д ля исп ользов ания в таких случ аях оп ератора **ANY** в место **EXISTS**.

**2.14. И спол ь зование COUNT вм есто EXISTS**

При отсутств ии **NULL**-знач ений оп ератор **EXISTS** может быть исп ользов ан в место **ANY** и **ALL**. Также в место **EXISTS** и **NOT** **EXISTS** могут быть исп ольз ов аны те же самые п од зап росы, но с исп ользов анием **COUNT**(\*) в п ред ложении **SELECT**. Н ап ример, з ап рос

**SELECT** \*

**FROM** UNIVERSITY A

**WHERE** **NOT** **EXISTS**

(**SELECT** \*

**FROM** UNIVERSITY B

**WHERE** A.RATING > = B.RATING  **AND** B.CITY = ‘М о сква ’);

может быть п ред став лен и в след ующ ем в ид е

**SELECT** \*

**FROM** UNIVERSITY A  **WHERE** 1 >

(**SELECT** **COUNT**(\*)

**FROM** UNIVERSITY B

**WHERE** A.RATING > = B.RATING  **AND** B.CITY = ‘М о сква ’); **УП РАЖ НЕ НИ Я**

1. Н ап ишите зап рос, в ыбира ющ ий д анные о наз в аниях унив ерситетов , рейтинг которых рав ен или п рев осход ит рейтинг Воронежского госуд арств енного унив ерситета.
2. Н ап ишите зап рос, исп ользующ ий **ANY** или **ALL**, в ып олняющ ий в ыборку д анных о студ ентах, у которых в город е их п остоянного местожительств а нет унив ерситета.
3. Н ап ишите зап рос, в ыбирающ ий из таблицы EXAM\_MARKS д анные о назв аниях п ред метов обуч ения, д ля которых знач ение п олуч енных на экзамене оценок (п оле MARK) п рев ышает любое знач ение оценки д ля п ред мета, имеющ его ид ентификатор рав ный 105. 40. Н ап ишите этот же з ап рос с исп ольз ов анием **MAX**.

**2.15. О ператор объединения UNION**

О п ератор **UNION** исп ользуется д ля объед инения в ыход ных д анных д в ух или более SQL-з ап росов в ед иное множеств о строк и столбцов . Н ап ример, д ля того, ч тобы п олуч ить в од ной таблице фамилии и ид ентификаторы студ ентов и п реп од ав ателей из М оскв ы, можно исп ользов ать след ующ ий зап рос.

**SELECT** ‘С туден т\_\_\_\_\_\_’, SURNAME,STUDENT\_ID

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’

## UNION

**SELECT** ‘Пр епо да ва тель’, SURNAME,LECTURER\_ID

**FROM** LECTURER

**WHERE** CITY = ‘М о сква ’;

О братите в нимание на то, ч то симв олом “;” (точ ка с зап ятой) оканч ив ается только п ослед ний з ап рос. О тсутств ие этого симв ола в конце **SELECT-**з ап роса означ ает, ч то след ующ ий з а ним з ап рос также, как и он сам, яв ляется ч астью общ его зап роса с **UNION**.

Исп ользов ание оп ератора **UNION** в оз можно только п ри объед инении

зап росов , соотв етств ующ ие столбцы которых *со вм ест и м ы по о бъ ед и нени ю*. То есть, соотв етств ующ ие ч ислов ые п оля д олжны иметь п олностью сов п ад ающ ие тип и размер, симв ольные п оля д олжны иметь точ но сов п ад ающ ее колич еств о симв олов . Если **NULL-**знач ения зап рещ ены д ля столбца хотя бы од ного любого п од зап роса объед инения, то они д олжны быть зап рещ ены и д ля в сех соотв етств ующ их столбцов в д ругих п од зап росах объед инения.

**2.16. Устранение дубл ирования в UNION**

В отлич ие от обыч ных зап росов **UNION** ав томатич ески исключ ает из в ыход ных д анных д убликаты строк, нап ример, в зап росе

**SELECT** CITY

**FROM** STUDENT

**UNION**

**SELECT** CITY

**FROM** LECTURER;

сов п ад ающ ие наименов ания город ов буд ут исключ ены.

Если в се же необход имо в кажд ом зап росе в ыв ести в се строки незав исимо от того, имеются ли такие же строки в д ругих объед иняемых зап росах, то след ует исп ользов ать в о множеств енном зап росе конструкцию с оп ератором **UNION** **ALL**. Так в з ап росе

**SELECT** CITY

**FROM** STUDENT

**UNION ALL**

**SELECT** CITY

**FROM** LECTURER;

д убликаты знач ений город ов , в ыв од имые в торой ч астью зап роса, не буд ут исключ аться.

Прив ед ем ещ е од ин п ример исп ользов ания оп ератора **UNION**. Пусть необход имо состав ить отч ет, сод ержащ ий д ля кажд ой д аты сд ач и экз аменов св ед ения п о кажд ому студ енту, п олуч ив шему максимальную или минимальную оценки.

**SELECT** ‘ма кс о ц’, A.STUDENT\_ID, SURNAME, MARK, EXAM\_DATE

**FROM** STUDENT A, EXAM\_MARKS B

**WHERE** (A.STUDENT\_ID = B.STUDENT\_ID

**AND** B.MARK =

(**SELECT** **MAX**(MARK)

**FROM** EXAM\_MARKS C

**WHERE** C.EXAM\_DATE = B.EXAM\_DATE))

**UNION ALL**

**SELECT** ‘ми н о ц ’, A.STUDENT\_ID, SURNAME, MARK, EXAM\_DATE **FROM** STUDENT A, EXAM\_MARKS B

**WHERE** (A.STUDENT\_ID = B.STUDENT\_ID

**AND** B.MARK =

(**SELECT** **MIN**(MARK)

**FROM** EXAM\_MARKS C

**WHERE** C.EXAM\_DATE = B.EXAM\_DATE));

Для отлич ия строк, в ыв од имых п ерв ой и в торой ч астями зап роса, в них в став лены текстов ые конста нты ‘ма кс о ц’ и ‘ми н о ц ’.

В п рив ед енном зап росе агрегирующ ие функции исп ользуются в п од зап росах. Это яв ляется нерациональным с точ ки з рения в ремени, затрач ив аемого на в ып олнение зап роса (см. разд ел 2.9). Б олее эффектив на форма з ап роса, в озв ращ ающ его аналогич ный результат:

**SELECT** ‘ма кс о ц’, A.STUDENT\_ID, SURNAME, E.MARK, E.EXAM\_DATE

**FROM** STUDENT A,

(**SELECT** B.STUDENT\_ID, B.MARK, B.EXAM\_DATE

**FROM** EXAM\_MARKS B,

(**SELECT MAX**(MARK) **AS** MAX\_MARK, C.EXAM\_DATE

**FROM** EXAM\_MARKS C

**GROUP** **BY** C.EXAM\_DATE)D

**WHERE** B.EXAM\_DATE=D.EXAM\_DATE **AND** B.MARK=MAX\_MARK) E

**WHERE** A.STUDENT\_ID=E.STUDENT\_ID

**UNION** **ALL**

**SELECT** ‘ми н о ц ’, A.STUDENT\_ID, SURNAME, E.MARK, E.EXAM\_DATE

**FROM** STUDENT A,

(**SELECT** B.STUDENT\_ID, B.MARK, B.EXAM\_DATE

**FROM** EXAM\_MARKS B,

(**SELECT MIN**(MARK) **AS** MIN\_MARK, C.EXAM\_DATE

**FROM** EXAM\_MARKS C

**GROUP** **BY** C.EXAM\_DATE) D

**WHERE** B.EXAM\_DATE=D.EXAM\_DATE

**AND** B.MARK=MIN\_MARK) E

**WHERE** A.STUDENT\_ID=E.STUDENT\_ID

**2.17. И спол ь зование UNION** **с ORDER** **BY**

Пред ложение **ORDER** **BY** п рименяется д ля уп оряд оч ения в ыход ных д анных объед инения зап росов так же, как и д ля отд ельных з ап росов . Послед ний п ример, п ри необход имости уп оряд оч ения в ыход ных д анных зап роса п о фамилиям студ ентов и д атам экзаменов , может в ыгляд еть так:

**SELECT** ‘ма кс о ц’, A.STUDENT\_ID, SURNAME, E.MARK, E.EXAM\_DATE

**FROM** STUDENT A,

(**SELECT** B.STUDENT\_ID, B.MARK, B.EXAM\_DATE

**FROM** EXAM\_MARKS B,

(**SELECT MAX**(MARK) **AS** MAX\_MARK, C.EXAM\_DATE

**FROM** EXAM\_MARKS C

**GROUP** **BY** C.EXAM\_DATE)D

**WHERE** B.EXAM\_DATE=D.EXAM\_DATE **AND** B.MARK=MAX\_MARK) E

**WHERE** A.STUDENT\_ID=E.STUDENT\_ID

**UNION ALL**

**SELECT** ‘ми н о ц ’, A.STUDENT\_ID, SURNAME, E.MARK, E.EXAM\_DATE

**FROM** STUDENT A,

(**SELECT** B.STUDENT\_ID, B.MARK, B.EXAM\_DATE

**FROM** EXAM\_MARKS B,

(**SELECT MIN**(MARK) **AS** MIN\_MARK, C.EXAM\_DATE **FROM** EXAM\_MARKS C

**GROUP BY** C.EXAM\_DATE) D

**WHERE** B.EXAM\_DATE=D.EXAM\_DATE **AND** B.MARK=MIN\_MARK) E

**WHERE** A.STUDENT\_ID=E.STUDENT\_ID **ORDER BY** SURNAME, E.EXAM\_DATE;

**2.18. В неш нее объ единение**

Часто п олезна оп ерация объед инения д в ух з ап росов , в которой в торой зап рос в ыбирает строки, исключ енные п ерв ым. Такая оп ерация назыв ается в нешним объед инением.

Рассмотрим п ример. Пусть в таблице STUDENT имеются з ап иси о студ ентах, в которых не указ ан ид ентификатор унив ерситета. Требуется состав ить сп исок студ ентов с указ анием наименов а ния унив ерситета д ля тех студ ентов , у которых эти д анные есть, но п ри этом не отбрасыв ая и студ ентов , у которых унив ерситет не указан. Можно п олуч ить желаемые св ед ения, сформиров ав объед инение д в ух зап росов , од ин из которых в ып олняет в ыборку студ ентов с наз в аниями их унив ерситетов , а в торой в ыбирает студ ентов с **NULL**-знач ениями в п оле UNIV\_ID. В д анном случ ае оказыв ается п олезной в озможность в став ки в з ап рос констант, в на шем случ ае текстов ой конста нты ‘н е и звестен ’, ч тобы отметить в сп иске тех студ ентов , у которых отсутств ует информация об унив ерситете.

**SELECT** SURNAME, NAME, UNIV\_NAME

**FROM** STUDENT, UNIVERSITY

**WHERE** STUDENT.UNIV\_ID = UNIVERSITY.UNIV\_ID **UNION**

**SELECT** SURNAME, NAME, ‘н е и звестен ’

**FROM** STUDENT

**WHERE** UNIV\_ID **IS** **NULL**

**ORDER** **BY** 1;

Для сов местимости столбцов объед иняемых з ап росов константу ‘н е и звестен ’ в о в тором з ап росе след ует доп олнить п робелами так, ч тобы ее д лина соотв етств ов ала д лине п оля UNIV\_NAME или исп ользов ать д ля согласов ания тип ов функцию **CAST**. В некоторых СУ Б Д согласов ание тип ов п оля и з амещ ающ ей его текстов ой константы осущ еств ляется ав томатич ески.

### УП РАЖ НЕ НИ Я

1. Соз д айте объед инение д в ух зап росов , которые в ыд ают з нач ения п олей UNIV\_NAME, CITY, RATING д ля в сех унив ерситетов . Те из них, у которых рейтинг рав ен или в ыше 300, д олжны иметь комментарий ‘Вы со ки й’, в се остальные – ‘Н и зки й’.
2. Н ап ишите кома нд у, которая в ыд ает сп исок фамилий студ ентов , с комментарием ‘успева ет’ у студ ентов , имеющ их в се п оложительные оценки, комментарием ‘н е успева ет’ д ля сд ав ав ших экз амены, но имеющ их хотя бы од ну неуд ов летв орительную оценку, и комментарием ‘н е сда ва л’ – д ля в сех остальных. В в ыв од имом результате фамилии студ ентов уп оряд оч ить п о алфав иту.
3. Выв ед ите объед иненный сп исок студ ентов и п реп од ав ателей, жив ущ их в М оскв е, с соотв етств ующ ими комментариями ‘студен т’ или

‘пр епо да ва тель’.

1. Выв ед ите объед иненный сп исок студ ентов и п реп од ав ателей Воронежского госуд арств енного унив ерситета с соотв етств ующ ими комментариями ‘студен т’ или ‘пр епо да ва тель’.

**2.19. С оединение табл иц с испол ь зованием оператора JOIN**

Если в оп ераторе **SELECT** п осле ключ ев ого слов а **FROM** указыв ается не од на, а д в е таблицы, то в рез ультате в ып олнения зап роса, в котором отсутств ует п ред ложение **WHERE**, кажд ая строка од ной таблицы буд ет соед инена с кажд ой строкой в торой таблицы. Такая оп ерация назыв ается *д ека р т о вым пр ои звед ени ем* или *по лны м* (**CROSS**) *со ед и нени ем* таблиц базы д анных. Сама п о себе эта оп ерация не имеет п рактич еского знач ения, более того, п ри ошибоч ном исп ользов ании она может п рив ести к неожид анным нештатным ситуациям, так как в этом случ ае в отв ете на з ап рос колич еств о зап исей буд ет рав но п роизв ед ению ч исла зап исей в соед иняемых таблица х, то есть может оказаться ч резв ыч айно большим. Соед инение таблиц имеет смысл тогд а, когд а соед иняются *не все* строки исход ных таблиц, а только те, которые интересуют п ользов ателя. Такое огранич ение может быть осущ еств лено с п омощ ью исп ользов а ния в з ап росе соотв етств ующ его услов ия в п ред ложении **WHERE**. Таким образом, SQL п озв оляет в ыв од ить информацию из нескольких таблиц, св языв ая их п о з нач ениям оп ред еленных п олей.

Н ап ример, если необход имо п олуч ить фамилии студ ентов (таблица STUDENT) и д ля кажд ого студ ента – наз в ания унив ерситетов (таблица UNIVERSITY), расп оложенных в город е, гд е жив ет студ ент, то необход имо п олуч ить в се комбинации з ап исей о студ ентах и унив ерситетах в обеих таблицах, в которых знач ение п оля CITY сов п ад ает. Это можно сд елать с п омощ ью след ующ его з ап роса.

**SELECT** STUDENT.SURNAME, UNIVERSITY.UNIV\_NAME, STUDENT.CITY

**FROM** STUDENT, UNIVERSITY

**WHERE** STUDENT.CITY = UNIVERSITY.CITY;

Соед инение, исп ользующ ее п ред икаты, основ анные на рав енств а х, назыв ается *экви со ед и нени ем* . Рассмотренный п ример соед инения таблиц относятся к в ид у так наз ыв аемого *внут р еннего* (**INNER**) *со ед и нени я*. При таком тип е соед инения соед иняются только те строки таблиц, д ля которых яв ляется истинным п ред икат, з ад ав аемый в п ред ложении **ON** в ып олняемого зап роса.

Прив ед енный в ыше зап рос может быть зап исан инач е, с исп ольз ов анием ключ ев ого слов а **JOIN**.

**SELECT** STUDENT.SURNAME, UNIVERSITY.UNIV\_NAME, STUDENT.CITY

**FROM** STUDENT **INNER** **JOIN** UNIVERSITY

**ON** STUDENT.CITY = UNIVERSITY.CITY;

К люч ев ое слов о **INNER** в зап росе может быть оп ущ ено, так как эта оп ция в оп ераторе **JOIN** д ейств ует п о умолч анию.

Рассмотренный в ыше случ ай п олного соед инения (д екартов а п роизв ед ения таблиц) с исп ользов анием ключ ев ого слов а **JOIN** буд ет в ыгляд еть след ующ им образом

**SELECT** \* **FROM** STUDENT **JOIN** UNIVERSITY;

ч то экв ив алентно

**SELECT** \* **FROM** STUDENT, UNIVERSITY;

Заметим, ч то в СУ Б Д Oracle зад ав аемый станд артом яз ыка SQL оп ератор **JOIN** не п од д ержив ается.

### 2.19.1. О перац ии соединения т абл иц посредст вом ссы л очной ц ел ост ност и

Информация в таблицах STUDENT и EXAM\_MARKS уже св яз ана п осред ств ом п оля STUDENT\_ID. В таблице STUDENT п оле STUDENT\_ID яв ляется п ерв ич ным ключ ом, а в таблице EXAM\_MARKS, ссылающ имся на него в нешним ключ ом. Состояние св язанных таким образом таблиц назыв ается состоянием ссылоч ной целостности. В д анном случ ае ссылоч ная целостность этих таблиц п од разумев ает, ч то *ка ж д о м у* з нач ению п оля STUDENT\_ID в таблице EXAM\_MARKS *о бяза т ельно* соотв етств ует *т а ко е ж е зна чени е* п оля STUDENT\_ID в таблице STUDENT. Другими слов ами, в таблице EXAM\_MARKS не может быть зап исей, имеющ их ид ентификаторы студ ентов , которых нет в таблице STUDENT. Станд артное п рименение оп ерации соед инения состоит в из в леч ении д анных в терминах этой св язи.

Чтобы п олуч ить сп исок фамилий студ ентов с п олуч енными ими оценками и ид ентификаторами п ред метов можно исп ользов ать след ующ ий зап рос:

**SELECT** SURNAME, MARK, SUBJ\_ID

**FROM** STUDENT, EXAM\_MARKS

**WHERE** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID;

Тот же самый результат может быть п олуч ен п ри исп ользов ании в зап росе д ля зад ания оп ерации соед инения таблиц ключ ев ого слов а **JOIN**. Зап рос с оп ератором **JOIN** в ыгляд ит след ующ им образом

**SELECT** SURNAME, MARK

**FROM** STUDENT **JOIN** EXAM\_MARKS

**ON** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID;

Х отя в ыше реч ь шла о соед инении д в ух таблиц, можно сформиров ать зап росы п утем соед инения более ч ем д в ух таблиц.

Пусть требуется найти фамилии в сех студ ентов , п олуч ив ших неуд ов летв орительную оценку, в месте с наз в аниями п ред метов обуч ения, п о которым п олуч ена эта оценка .

**SELECT** SUBJ\_NAME, SURNAME, MARK

**FROM** STUDENT, SUBJECT, EXAM\_MARKS

**WHERE** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID

**AND** SUBJECT.SUBJ\_ID = EXAM\_MARKS.SUBJ\_ID

**AND** EXAM\_MARKS.MARK = 2;

То же самое с исп ользов а нием оп ератора **JOIN**

**SELECT** SUBJ\_NAME, SURNAME, MARK

**FROM** STUDENT **JOIN** SUBJECT **JOIN** EXAM\_MARKS **ON** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID

**AND** SUBJECT.SUBJ\_ID = EXAM\_MARKS.SUBJ\_ID  **AND** EXAM\_MARKS.MARK = 2;

### 2.19.2. В неш нее соединение т абл иц

К ак отмеч алось ранее, п ри исп ользов ании *внут р еннего* (**INNER**) соед инения таблиц соед иняются только те их строки, в которых сов п ад ают знач ения п олей, зад ав аемые в п ред ложении **WHERE** зап роса. О д нако в о многих случ аях это может п рив ести к нежелательной п отере информации. Рассмотрим ещ е раз п рив ед енный в ыше п ример зап роса на в ыборку сп иска фамилий студ ентов с п олуч енными ими оценками и ид ентификаторами п ред метов . При исп ольз ов ании, как это было сд елано в рассматрив аемом п римере, в нутреннего соед инения в результат зап роса не п оп ад ут студ енты, которые ещ е не сд ав али экзамены и которые, след ов ательно, отсутств уют в таблице EXAM\_MARKS. Если же необход имо иметь зап иси об этих студ ентах в в ыд ав аемом зап росом сп иске, то можно п рисоед инить св ед ения о студ ентах, не сд ав ав ших экз амен, п утем исп ользов ания оп ератора **UNION** с соотв етств ующ им зап росом. Н ап ример, след ующ им образом:

**SELECT** SURNAME, **CAST** MARK **AS** CHAR(1), **CAST** SUBJ\_ID **AS** CHAR(10)

**FROM** STUDENT, EXAM\_MARKS

**WHERE** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID  **UNION**

**SELECT** SURNAME, **CAST** **NULL** **AS** CHAR(1), **CAST** **NULL** **AS** CHAR(10)

**FROM** STUDENT

**WHERE NOT EXIST**

(**SELECT** \*

**FROM** EXAM\_MARKS

**WHERE** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID);

(зд есь функция п реобразов ания тип ов **CAST** исп ольз уется д ля обесп еч ения сов местимости тип ов п олей объед иняемых зап росов ).

Н ужный результат, од нако, может быть п олуч ен и п утем исп ользов ания *внеш него со ед и нени я*, точ нее од ной из его разнов ид ностей – *левого внеш него со ед и нени я*, с исп ользов анием которого зап рос буд ет в ыгляд еть след ующ им образ ом:

**SELECT** SURNAME, MARK

**FROM** STUDENT **LEFT** **OUTER** **JOIN** EXAM\_MARKS

**ON** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID;

При исп ользов ании *лево го* соед инения расширение в ыв од имой таблицы осущ еств ляется з а сч ет зап исей в ход ной таблицы, имя которой указано *слева* от оп ератора **JOIN**.

След ует з аметить, ч то нотация з ап росов с в нешним соед инением в СУ Б Д

ORACLE отлич ается от п рив ед енной нотации, зад ав аемой станд артом языка SQL. В нотации, исп ольз уемой в Oracle, этот же зап рос буд ет иметь в ид

**SELECT** SURNAME, MARK, SUBJ\_ID

**FROM** STUDENT, EXAM\_MARKS

**WHERE** STUDENT.STUDENT\_ID = EXAM\_MARKS.STUDENT\_ID(**+**);

Знак (**+**) став ится у той таблицы, которая д оп олняется зап исями с **NULL**знач ениями, ч тобы п ри соед инении таблиц в в ыход ное отношение п оп а ли и те зап иси д ругой таблицы, д ля которых в таблице со знаком (**+**) не наход ится строк с соотв етств ующ ими знач ениями атрибутов , исп ользуемых д ля соед инения. То есть д ля *лево го* в нешнего соед инения (п о нотации станд арта SQL) в зап росе ORACLE-SQL указатель (**+**) став ится у *пр а во й* таблицы.

Прив ед енный в ыше зап рос может быть реализов ан и с п рименением *пр а вого* *внеш него со ед и нени я*. О н буд ет иметь след ующ ий в ид

**SELECT** SURNAME, MARK

**FROM** EXAM\_MARKS **RIGHT** **OUTER** **JOIN** STUDENT

**ON** EXAM\_MARKS.STUDENT\_ID = STUDENT.STUDENT\_ID;

Зд есь таблица STUDENT, за сч ет з ап исей которой осущ еств ляется расширение в ыв од имой таблицы, стоит сп рав а от оп ератора **JOIN**.

В нотации Oracle этот зап рос буд ет в ыгляд еть след ующ им образом.

**SELECT** SURNAME, MARK, SUBJ\_ID

**FROM** STUDENT, EXAM\_MARKS

**WHERE** EXAM\_MARKS.STUDENT\_ID(**+**) = STUDENT.STUDENT\_ID;

Вид но, ч то исп ольз ов ание в нешнего п рав ого или лев ого соед инения

п озв оляет сущ еств енно уп ростить зап рос, сд елать его зап ись более комп актной.

Иногд а в озникает необход имость в ключ ения в результат зап роса зап исей из обеих (п рав ой и лев ой) соед иняемых таблиц, д ля которых не уд ов летв оряется услов ие соед инения. Такое соед инение наз ыв ается *по лны м внеш ни м со ед и нени ем* и осущ еств ляется указ анием в за п росе ключ ев ых слов **FULL OUTER JOIN** или **UNION JOIN**.

#### УП РАЖ НЕ НИ Я

1. Н ап ишите з ап рос, который в ып олняет в ыв од д анных о фамилиях, *сд а ва вш и х* экзамены студ ентов , в месте с ид ентификаторами кажд ого сд анного ими п ред мета обуч ения.
2. Н ап ишите зап рос, который в ып олняет в ыборку знач ений фамилии *всех* студ ентов с указанием д ля студ ентов , сд ав ав ших экзамены, ид ентификаторов сд анных ими п ред метов обуч ения.
3. Н ап ишите з ап рос, который в ып олняет в ыв од д анных о фамилиях студ ентов , *сд а ва вш и х* экзамены, в месте с наименов аниями кажд ого сд анного ими п ред мета обуч ения.
4. Н ап ишите з ап рос на в ыд ач у д ля кажд ого студ ента назв аний в сех п ред метов обуч ения, п о которым этот студ ент п олуч ил оценку 4 или 5.
5. Н ап ишите зап рос на в ыд ач у д анных о назв аниях в сех п ред метов , п о которым студ енты п олуч или только хорошие (4 и 5) оценки. В в ыход ных д анных д олжны быть п рив ед ены фамилии студ ентов , наз в ания п ред метов и оценка.
6. Н ап ишите зап рос, который в ып олняет в ыв од сп иска унив ерситетов с рейтингом, п рев ышающ им 300, в месте со знач ением максимального размера стип енд ии, п олуч аемой студ ентами в этих унив ерситетах.
7. Н ап ишите зап рос на в ыд ач у сп иска фамилий студ ентов (в алфав итном п оряд ке) в месте со з нач ением рейтинга унив ерситета, гд е кажд ый из них уч ится, в ключ ив в сп исок и тех студ ентов , д ля которых в базе д анных не указано место их уч ебы.

### 2.19.3. И спол ь з ование псевдоним ов при соединении т абл иц

Часто п ри п олуч ении информации из таблиц базы д анных необход имо осущ еств лять соед инение таблицы с ее же коп ией. Н ап ример, это требуется в случ ае, когд а требуется найти фамилии студ ентов , имеющ их од инаков ые имена. При соед инении таблицы с ее же коп ией в в од ят п сев д онимы (алиасы) таблицы. Зап рос д ля п оиска фамилий студ ентов , имеющ их од инаков ые имена, в ыгляд ит след ующ им образом

**SELECT** FIRST.SURNAME, SECOND.SURNAME

**FROM** STUDENT FIRST, STUDENT SECOND

**WHERE** FIRST.NAME = SECOND.NAME

В этом зап росе в в ед ены д в а п сев д онима д ля од ной таблицы STUDENT, ч то п озв оляет корректно зад ать в ыражение, св языв ающ ее д в е коп ии таблицы. Чтобы исключ ить п ов торения строк в в ыв од имом результате зап роса из-за п ов торного срав нения од ной и той же п ары студ ентов , необход имо зад ать п оряд ок след ов ания д ля д в ух з нач ений так, ч тобы од но знач ение было меньше, ч ем д ругое, ч то д елает п ред икат асимметрич ным.

**SELECT** FIRST.SURNAME, SECOND.SURNAME

**FROM** STUDENT FIRST, STUDENT SECOND

**WHERE** FIRST.NAME = SECOND.NAME

**AND** FIRST.SURNAME < SECOND.SURNAME

#### УП РАЖ НЕ НИ Я

1. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска в сех п ар фамилий студ ентов , п рожив а ющ их в од ном город е. При этом не в ключ ать в сп исок комбинации фамилий студ ентов самих с собой (то есть комбинацию тип а “Ив анов -Ив анов ”) и комбинации фамилий студ ентов , отлич ающ иеся п оряд ком след ов ания (то есть в ключ ать од ну из д в ух комбинаций тип а “Ив анов -Петров ” и “Петров -Ив анов ”).
2. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска в сех п ар наз в аний унив ерситетов , расп оложенных в од ном город е, не в ключ ая в сп исок комбинации назв аний унив ерситетов самих с собой и п ары наз в аний унив ерситетов , отлич ающ иеся п оряд ком след ов а ния.
3. зап рос, который п озв оляет п олуч ить д анные о назв аниях унив ерситетов и городов , в которых они расп оложены, с рейтингом, рав ным или п рев ышающ им рейтинг ВГ У .

#### ДО П О ЛНИ Т ЕЛЬНЫ Е УП РАЖ НЕ НИ Я НА В Ы БО РКУ Д АННЫ Х

1. Н ап исать зап рос, в ып олняющ ий в ыв од д анных об именах и фамилиях студ ентов , п олуч ив ших хотя бы од ну отлич ную оценку.
2. Н ап исать зап рос, в ып олняющ ий в ыв од д анных об именах и фамилиях студ ентов , имеющ их в есь набор оценок (тройки, ч етв ерки и п ятерки).
3. Н ап исать зап рос, в ып олняющ ий в ыборку знач ений ид ентификаторов студ ентов , имеющ их такие же оценки, ч то и студ ент с ид ентификатором

12.

1. Н ап исать зап рос, в ып олняющ ий в ыборку в сех п ар ид ентификаторов п реп од ав ателей, в ед ущ их од инаков ые п ред меты обуч ения.
2. Н ап исать зап рос, в ып олняющ ий в ыв од д анных об именах и фамилиях студ ентов , не п олуч ив ших ни од ной отлич ной оценки.
3. Н ап исать зап рос, в ып олняющ ий в ыборку знач ений наименов аний п ред метов обуч ения, на п реп од ав а ние которых отв од ится более 50 ч асов .
4. Н ап исать зап рос, в ып олняющ ий в ыв од колич еств а студ ентов , не имеющ их ни од ной оценки.
5. Н ап исать зап рос, в ып олняющ ий в ыв од колич еств а студ ентов , имеющ их только отлич ные оценки.
6. Н ап исать з ап рос, в ып олняющ ий в ыв од д анных о п ред метах обуч ения, которые п реп од аются п реп од ав ателем п о фамилии К олесников .
7. Н ап исать зап рос, в ып олняющ ий в ыв од имен и фамилий п реп од ав ателей, п ров од ящ их занятия на п ерв ом курсе.
8. Н ап исать зап рос, в ып олняющ ий в ыв од имен и фамилий студ ентов , место п рожив ания которых не сов п ад ает с город ом, в котором наход ится их унив ерситет.
9. Н ап исать зап рос, в ып олняющ ий в ыв од колич еств а экзаменов , сд анных (с п оложительной оценкой) студ ентом с ид ентификатором 32.
10. зап рос, в ып олняющ ий в ыв од имен и фамилий п реп од ав ателей, ч итающ их д в а и более различ ных п ред мета обуч ения.
11. Н ап исать зап рос, в ып олняющ ий в ыв од имен и фамилий п реп од ав ателей, п ров од ящ их занятия в д в ух и более семестрах.
12. Н ап исать зап рос, в ып олняющ ий в ыв од д анных о наименов аниях п ред метов обуч ения, ч итаемых д в умя и более п реп од ав ателями.
13. Н ап исать зап рос, в ып олняющ ий в ыв од д ля кажд ого п ред мета обуч ения, п реп од ав аемого д ля студ ентов ВГ У , его наименов ание, фамилию и имя п реп од ав ателя, и город , в котором жив ет студ ент.
14. Н ап исать зап рос, в ып олняющ ий в ыв од колич еств а ч асов занятий, п ров од имых п реп од ав ателем Лагутиным.
15. Н ап исать зап рос, в ып олняющ ий в ыв од фамилий п реп од ав ателей, ч итающ их такие же п ред меты обуч ения, ч то и п реп од ав атель Сорокин.
16. Н ап исать зап рос, в ып олняющ ий в ыв од фамилий п реп од ав ателей, уч ебная нагруз ка которых (колич еств о уч ебных ч асов ) п рев ышает нагрузку п реп од ав ателя Н иколаев а.
17. Н ап исать зап рос, в ып олняющ ий в ыв од д анных о п реп од ав ателях, в ед ущ их обуч ение хотя бы п о од ному из п ред метов обуч ения, которые п реп од аются п реп од ав ателем п о фамилии Сорокин.
18. Н ап исать зап рос, в ып олняющ ий в ыв од д анных о фамилиях п реп од ав ателей, п реп од ающ их студ ентам, обуч ающ имся в унив ерситетах с рейтингом, меньшим 200.
19. Н ап исать зап рос, в ып олняющ ий в ыв од д анных о наименов аниях унив ерситетов , расп оложенных в Москв е, и имеющ их рейтинг меньше, ч ем у ВГ У .
20. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска фамилий студ ентов , обуч аемых в унив ерситете, расп оложенном в город е, назв ание которого стоит п ерв ым в алфав итном сп иске город ов .
21. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска студ ентов , сред няя оценка которых п рев ышает 4 балла.
22. Н ап исать зап рос, в ып олняющ ий в ыв од общ его колич еств а уч ебных ч асов з анятий, п ров од имых д ля студ ентов п ерв ого курса ВГ У .
23. з ап рос, в ып олняющ ий в ыв од сред него колич еств а уч ебных ч асов п ред метов обуч ения, п реп од ав аемых студ ентам в торого курса ВГ У .
24. Н ап исать зап рос, в ып олняющ ий в ыв од колич еств а студ ентов , имеющ их хотя бы од ну неуд ов летв орительную оценку и п рожив ающ их в город е, не сов п ад ающ ем с город ом их унив ерситета.
25. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска фамилий студ ентов , имеющ их только отлич ные оценки и п рожив ающ их в город е, не сов п ад ающ ем с город ом их унив ерситета.
26. Н ап исать зап рос, в ып олняющ ий в ыв од сп иска фамилий студ ентов , имеющ их д в е и более отлич ных оценки в кажд ом семестре, и п рожив ающ их в город е, не сов п ад ающ ем с город ом их унив ерситета.
27. Прив ед ите как можно больше формулиров ок зап роса “Получ ить фамилии студ ентов , сд а в ав ших экзамен п о информатике”.
28. Прив ед ите как можно больше формулиров ок зап роса “Получ ить фамилии п реп од ав ателей, п реп од ающ их информатику”.

1. М а н и пули р о ва н и е да н н ы ми

**3.1. Ком анды м анипул ирования данны м и**

В SQL д ля в ып олнения оп ераций в в од а д анных в таблицу, их из менения и уд аления п ред наз нач ены три команд ы языка манип улиров а ния д анными (DML). Это команд ы – **INSERT** (вставить), **UPDATE** (обновить), **DELETE**

(удалить).

К оманд а **INSERT** осущ еств ляет **в ста в ку** в таблицу нов ой строки. В п ростейшем случ ае она имеет след ующ ий в ид :

**INSERT** **INTO** <*и м я т а бли цы* > **VALUES** (<*зна чени е*>, <*зна чени е*>, … );

При такой зап иси указанные в скобках п осле ключ ев ого слов а **VALUES** знач ения в в од ятся в п оля д обав ленной в таблицу нов ой строки в том п оряд ке, в котором соотв етств ующ ие столбцы указ аны п ри созд ании таблицы, то есть в оп ераторе **CREATE** **TABLE**.

Н ап ример, в в од нов ой строки в таблицу STUDENT может быть осущ еств лен след ующ им образом

**INSERT** **INTO** STUDENT

**VALUES** (101,‘Ива н о в’,‘Алекса н др ’, 200, 3,‘М о сква ’,‘6/10/1979’, 15);

Чтобы такая команд а могла быть в ып олнена , таблица с указ анным в ней именем (STUDENT) д олжна быть п ред в арительно оп ред елена (созд ана) команд ой **CREATE** **TABLE**. Если в какое-либо п оле необход имо в став ить **NULL**-знач ение, то оно в в од ится как обыч ное з нач ение:

**INSERT** **INTO** STUDENT

**VALUES** (101,‘Ива н о в’, **NULL**, 200, 3,‘М о сква ’,‘6/10/1979’, 15);

В случ аях, когд а необход имо в в ести знач ения п олей в п оряд ке, отлич ном от п оряд ка столбцов , зад анного команд ой **CREATE** **TABLE**, или если требуется в в ести з нач ения не в о в се столбцы, то след ует исп ользов ать след ующ ую форму команд ы **INSERT**:

**INSERT** **INTO** STUDENT (STUDENT\_ID, CITY, SURNAME, NAME)  **VALUES** (101, ‘М о сква ’, ‘Ива н о в’, ‘С а ша ’);

Столбцам, наименов ания которых не указаны в п рив ед енном в скобках сп иске, ав томатич ески п рисв аив ается з нач ение п о умолч анию, если оно назнач ено п ри оп исании таблицы (команд а **CREATE** **TABLE)**, либо знач ение **NULL**.

С п омощ ью команд ы **INSERT** можно изв леч ь знач ение из од ной таблицы и разместить его в д ругой, к п римеру, з ап росом след ующ его в ид а:

**INSERT** **INTO** STUDENT1

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

При этом таблица STUDENT1 д олжна быть п ред в арительно соз д ана команд ой **CREATE** **TABLE** (раз д ел 4.1) и иметь структуру, ид ентич ную таблице STUDENT.

***У да ление*** строк из таблицы осущ еств ляется с п омощ ью команд ы **DELETE**.

След ующ ее в ыражение уд аляет в се строки таблицы EXAM\_MARKS1.

**DELETE** **FROM** EXAM\_MARKS1;

В результате таблица ста нов ится п устой (п осле этого она может быть уд алена команд ой **DROP** **TABLE**).

Для уд аления из таблицы сраз у нескольких строк, уд ов летв оряющ их некоторому услов ию можно в осп ользов аться п ред ложением **WHERE**, нап ример,

**DELETE** **FROM** EXAM\_MARKS1  **WHERE** STUDENT\_ID = 103;

М ожно уд алить груп п у строк

**DELETE** **FROM** STUDENT1  **WHERE** CITY = ‘М о сква ’; К оманд а **UPDATE** п оз в оляет **изменять**, то есть обнов лять, знач ения некоторых или в сех п олей в сущ еств ующ ей строке или строках таблицы. Н ап ример, ч тобы д ля в сех унив ерситетов , св ед ения о которых наход ятся в таблице UNIVERSITY1, изменить рейтинг на з нач ение 200, можно исп ользов ать конструкцию:

**UPDATE** UNIVERSITY1

**SET** RATING = 200;

Для указания конкретных строк таблицы, знач ения п олей которых д олжны быть изменены, в команд е **UPDATE** можно исп ользов ать п ред икат, указ ыв аемый в п ред ложении **WHERE**.

**UPDATE** UNIVERSITY1

**SET** RATING = 200

**WHERE** CITY = ‘М о сква ’;

В результате в ып олнения этого зап роса буд ет изменен рейтинг только у унив ерситетов , расп оложенных в М оскв е.

К оманд а **UPDATE** п оз в оляет изменять не только од ин, но и множеств о столбцов . Для указ ания конкретных столбцов , знач ения которых д олжны быть мод ифициров а ны, исп ольз уется п ред ложение **SET**.

Н ап ример, наименов ание п ред мета обуч ения ‘М а тема ти ка’ (д ля него SUBJ\_ID = 43) д олжно быть заменено, на наз в ание ‘Вы сша я ма тема ти ка’, п ри этом ид ентификационный номер необход имо сохранить, но в соотв етств ующ ие п оля строки таблицы в в ести нов ые д анные об этом п ред мете обуч ения. Зап рос буд ет в ыгляд еть след ующ им образом.

**UPDATE** SUBJECT1

**SET** SUBJ\_NAME = ‘Вы сша я ма тема ти ка’, HOUR = 36, SEMESTER

= 1

**WHERE** SUBJ\_ID = 43;

В п ред ложении **SET** команд ы **UPDATE** можно исп ользов ать скалярные в ыражения, указыв ающ ие сп особ изменения знач ений п оля, в которые могут в ход ить з нач ения из меняемого и д ругих п олей.

**UPDATE** UNIVERSITY1

**SET** RATING = RATING\*2;

Н ап ример, д ля ув елич ения в таблице STUDENT1 з нач ения п оля STIPEND в д в а раза д ля студ ентов из М оскв ы можно исп ользов ать зап рос

**UPDATE** STUDENT1

**SET** STIPEND = STIPEND\*2

**WHERE** CITY = ‘М о сква ’;

Пред ложение **SET** не яв ляется п ред икатом, п оэтому в нем можно указать знач ение **NULL** след ующ им образом.

**UPDATE** UNIVERSITY1

**SET** RATING = **NULL**

**WHERE** CITY = ‘М о сква ’;

**УП РАЖ НЕ НИ Я**

1. Н ап ишите команд у, которая в в од ит в таблицу SUBJECT строку д ля нов ого п ред мета обуч ения со след ующ ими з нач ениями п олей:

SEMESTER = 4; SUBJ\_NAME = ‘Алгеб р а ’; HOUR = 72; SUBJ\_ID = 201.

1. Вв ед ите з ап ись д ля нов ого студ ента, которого зов ут О рлов Н иколай, обуч ающ егося на п ерв ом курсе ВГ У , жив ущ его в Воронеже, св ед ения о д ате рожд ения и размере стип енд ии не из в естны.
2. Н ап ишите команд у, уд аляющ ую из таблицы EXAM\_MARKS зап иси обо в сех оценка х студ ента, ид ентификатор которого рав ен 100.
3. Н ап ишите команд у, которая ув елич ив ает на 5 знач ение рейтинга в сех, имеющ ихся в базе д анных унив ерситетов , расп оложенных в СанктПетербурге.
4. Измените в таблице знач ение город а, в котором п рожив ает студ ент Ив анов , на “Во р о н еж ”.

**3.2. И спол ь зование подзапросов в INSERT**

Применение оп ератора **INSERT** с п од з ап росом п оз в оляет загружать сразу несколько строк в од ну таблицу, исп ользуя информацию из д ругой таблицы. В то в ремя как оп ератор **INSERT**,использующий **VALUES** д обав ляет только од ну строку, **INSERT** с п од з ап росом д обав ляет в таблицу столько строк, сколько п од з ап рос изв лекает из д ругой таблицы. При этом колич еств о и тип в оз в ращ аемых п од зап росом столбцов д олжно соотв етств ов ать колич еств у и тип у столбцов таблицы, в которую в став ляются д анные.

Н ап ример, п усть таблица STUDENT1 имеет структуру, п олностью сов п ад ающ ую со структурой таблицы STUDENT. Зап рос, п озв оляющ ий зап олнить таблицу STUDENT1 зап исями из таблицы STUDENT обо в сех студ ентах из Москв ы, в ыгляд ит след ующ им образом.

**INSERT** **INTO** STUDENT1

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

Для того же, ч тобы добав ить в таблицу STUDENT1 св ед ения обо в сех студ ентах, которые *уча т ся* в М оскв е, можно исп ольз ов ать в п ред ложении

**WHERE** соотв етств ующ ий п од з ап рос. Н а п ример,

**INSERT** **INTO** STUDENT1

**SELECT** \*

**FROM** STUDENT

**WHERE** UNIV\_ID **IN**

(**SELECT** UNIV\_ID

**FROM** UNIVERSITY

**WHERE** CITY = ‘М о сква ’);

**3.2.1. И спол ь з ование подз апросов, основанны х на т абл иц ах внеш них з апросов**

Пред п оложим, сущ еств ует таблица SSTUD, в которой хранятся св ед ения о студ ентах, обуч ающ ихся в том же город е, в котором они жив ут. Можно зап олнить эту таблицу д анными из таблицы STUDENT, исп ользуя св язанные п од зап росы, след ующ им образ ом:

**INSERT** **INTO** SSTUD

**SELECT** \*

**FROM** STUDENT A

**WHERE** CITY **IN**

(**SELECT** CITY

**FROM** UNIVERSITY B

**WHERE** A.UNIV\_ID = B.UNIV\_ID);

Пред п оложим, ч то требуется в ыбрать сп исок студ ентов , имеющ их максимальный балл на кажд ый д ень сд ач и экз аменов , и разместить его в д ругой таблице с именем EXAM. Это можно осущ еств ить с п омощ ью зап роса

**INSERT** **INTO** EXAM

**SELECT** EXAM\_ID, STUDENT\_ID, SUBJ\_ID, MARK, EXAM\_DATE

**FROM** EXAM\_MARKS A  **WHERE** MARK =

**(SELECT MAX**(MARK)  **FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE);

**3.2.2. И спол ь з ование подз апросов с DELETE**

Пусть филиал унив ерситета в Н ью-Васюках ликв ид иров ан, и требуется уд алить из таблицы STUDENT з ап иси о студ ентах, которые там уч ились. Эту оп ерацию можно в ып олнить с п омощ ью зап роса

**DELETE**

**FROM** STUDENT

**WHERE** UNIV\_ID **IN**

(**SELECT** UNIV\_ID

**FROM** UNIVERSITY

**WHERE** CITY = ‘Н ью-Ва сюки’);

В п ред икате п ред ложения **FROM** (п од з ап роса) нельз я ссылаться на таблицу, из которой осущ еств ляется уд аление. О д нако можно ссылаться на текущ ую строку из таблицы, яв ляющ уюся канд ид атом на уд аление, то есть на строку, которая в настоящ ее в ремя п ров еряется в основ ном п ред икате.

**DELETE**

**FROM** STUDENT

**WHERE** EXISTS (**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING = 401

**AND** STUDENT.UNIV\_ID = UNIVERSITY.UNIV\_ID);

Часть **AND** п ред иката в нутреннего з ап роса ссылается на таблицу STUDENT. К оманд а уд аляет д анные о студ ентах, которые уч атся в унив ерситетах с рейтингом рав ным 401. Сущ еств уют и д ругие сп особы решения этой зад ач и.

**DELETE**

**FROM** STUDENT

**WHERE** 401 **IN**

(**SELECT** RATING

**FROM** UNIVERSITY

**WHERE** STUDENT.UNIV\_ID = UNIVERSITY.UNIV\_ID);

Пусть нужно найти наименьшее з нач ение оценки, п олуч енной в кажд ый д ень сд ач и экзаменов , и уд алить из таблицы св ед ения о студ енте, который п олуч ил эту оценку. Зап рос буд ет иметь в ид

**DELETE**

**FROM** STUDENT

**WHERE** STUDENT\_ID **IN**

(**SELECT** STUDENT\_ID  **FROM** EXAM\_MARKS A  **WHERE** MARK=

(**SELECT** **MIN**(MARK)

**FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE));

Так как столбец STUDENT\_ID яв ляется п ерв ич ным ключ ом, то уд аляется ед инств енна я строка.

Если в какой-то д ень сд ав ался только од ин экз амен (то есть, п олуч ена только од на минимальная оценка), и п о какой-либо п рич ине зап ись, в которой наход ится эта оценка , требуется остав ить, то решение буд ет иметь в ид :

**DELETE**

**FROM** STUDENT

**WHERE** STUDENT\_ID **IN**

(**SELECT** STUDENT\_ID  **FROM** EXAM\_MARKS A

**WHERE** MARK =

(**SELECT** **MIN**(MARK)

**FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE

**AND** 1 <

(**SELECT** **COUNT**(SUBJ\_ID)

**FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE)));

**3.2.3. И спол ь з ование подз апросов с UPDATE**

С п омощ ью команд ы **UPDATE** можно п рименять п од зап росы в любой форме, п риемлемой д ля команд ы **DELETE**.

Н ап ример, исп ользуя св яз анные п од зап росы, можно ув елич ить знач ение размера стип енд ии на 20 в з ап исях студ ентов , сд ав ших экз амены на 4 и 5.

**UPDATE** STUDENT1

**SET** STIPEND = STIPEND + 20

**WHERE** 4 <=

(**SELECT** **MIN**(MARK)

**FROM** EXAM\_MARKS

**WHERE** EXAM\_MARKS.STUDENT\_ID =

STUDENT1.STUDENT\_ID);

Другой зап рос: “У меньшить в елич ину стип енд ии на 20 в сем студ ентам, п олуч ив шим на экзамене минима льную оценку”.

**UPDATE** STUDENT1

**SET** STIPEND = STIPEND – 20

**WHERE** STUDENT\_ID **IN**

(**SELECT** STUDENT\_ID

**FROM** EXAM\_MARKS A

**WHERE** MARK =

(**SELECT** **MIN**(MARK)

**FROM** EXAM\_MARKS B

**WHERE** A.EXAM\_DATE = B.EXAM\_DATE));

### УП РАЖ НЕ НИ Я

1. Пусть сущ еств ует таблица с именем STUDENT1, оп ред еления столбцов которой п олностью сов п ад ают с оп ред елениями столбцов таблицы STUDENT. Встав ить в эту таблицу св ед ения о студ ентах, усп ешно сд ав ших экзамены более ч ем п о п яти п ред метам обуч ения.
2. Н ап ишите команд у, уд аляющ ую из таблицы SUBJECT1 св ед ения о п ред метах обуч ения, п о которым студ ентами не п олуч ено ни од ной оценки.
3. Н ап ишите зап рос, ув елич ив ающ ий д анные о в елич ине стип енд ии на 20% в сем студ ентам, у которых общ ая сумма баллов п рев ышает знач ение 50.

4. С о зда н и е о б ъ екто в б а зы да н н ы х **4.1. С оздание табл иц базы данны х**

Созд ание объектов базы д анных осущ еств ляется с п омощ ью оп ераторов языка оп ред еления д анных (DDL).

Таблицы баз ы д анных созд аются с п омощ ью команд ы **CREATE** **TABLE**. Эта команд а созд ает п устую таблицу, то есть таблицу, не имеющ ую строк. Знач ения в эту таблицу в в од ятся с п омощ ью команд ы **INSERT**. К оманд а **CREATE** **TABLE** оп ред еляет имя таблицы и множеств о п оименов анных столбцов в указанном п оряд ке. Для кажд ого столбца д олжен быть оп ред елен тип и раз мер. К ажд ая созд ав аемая таблица д олжна иметь, п о крайней мере, од ин столбец. Синтаксис команд ы **CREATE** **TABLE** имеет след ующ ий в ид .

**CREATE** **TABLE** <*и м я т а бли цы* >

(<*и м я ст о лбца* ><*т и п д а нных*>[(<*р а зм ер* >)],… );

Исп ользуемые в SQL тип ы д анных, как минимум, п од д ержив ают станд арты ANSI (*American National Standards Institute – А м ер и ка нски й на ци о на льны й и нст и т ут ст а нд а р т ов*) (см. разд ел 1.5.Тип ы д анных SQL):

**CHAR**(**CHARACTER**),

**INT** (**INTEGER**),

**SMALLINT**,

**DEC** (**DECIMAL**),

**NUMERIC**, **FLOAT**, … .

Тип д анных, д ля которого обяз ательно д олжен быть указан размер – это **CHAR**. Реальное колич еств о симв олов , которое может наход иться в п оле, меняется от нуля (если в п оле сод ержится **NULL**-знач ение) д о з ад анного в **CREATE** **TABLE** максимального з нач ения.

След ующ ий п ример п оказыв ает команд у, которая п оз в оляет созд ать таблицу STUDENT.

**CREATE** **TABLE** STUDENT1

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER**, |
| SURNAME | **VARCHAR(60)**, |
| NAME | **VARCHAR(60)**, |
| STIPEND | **DOUBLE**, |
| KURS | **INTEGER**, |
| CITY | **VARCHAR(60)**, |
| BIRTHDAY | **DATE**, |
| UNIV\_ID | **INTEGER**); |

**4.2. И спол ь зование индексац ии дл я бы строго дост упа к данны м**

О п ерации п оиска-в ыборки (**SELECT**) д анных из таблиц п о з нач ениям их п олей могут быть сущ еств енно ускорены п утем исп ользов ания инд ексации д анных. Инд екс сод ержит уп оряд оч енный (в а лфав итном или ч ислов ом п оряд ке) сп исок сод ержимого столбцов или груп п ы столбцов в инд ексируемой таблице с ид ентификаторами этих строк (**ROWID**). Д ля п ольз ов ателей инд ексиров ание таблицы п о тем или иным столбцам п ред став ляет собой сп особ *ло ги ческо го* уп оряд оч ив ания знач ений инд ексиров анных столбцов , п озв оляющ его, в отлич ие от п ослед ов ательного п еребора строк, сущ еств енно п ов ысить скорость доступ а к конкретным строкам таблицы п ри в ыборках, исп ольз ующ их знач ения этих столбцов . Инд ексация п оз в оляет наход ить сод ержащ ий инд ексиров анную строку блок д анных, в ып олняя небольшое ч исло обращ ений к в нешнему устройств у хранения д анных.

При исп ользов ании инд ексации след ует, од нако, иметь в в ид у, ч то уп рав ление инд ексом сущ еств енно замед ляет в ремя в ып олнения оп ера ций, св яз анных с обнов лением д анных (таких, как **INSERT** и **DELETE**), так как эти оп ерации требуют п ерестройки инд ексов .

Инд ексы можно соз д ав ать как п о од ному, так и п о множеств у п олей. Если указано более од ного п оля д ля созд ания ед инств енного инд екса, д анные уп оряд оч ив аются п о з нач ениям п ерв ого п оля, п о которому осущ еств ляется инд ексиров ание. Внутри п олуч ив шейся груп п ы осущ еств ляется уп оряд оч ив ание п о з нач ениям в торого п оля, д ля п олуч ив шихся в результате груп п осущ еств ляется уп оряд оч ив ание п о знач ениям третьего п оля и т.д .

Синтаксис команд ы созд ания инд екса имеет след ующ ий в ид :

**CREATE** **INDEX** <*и м я и нд екса* > **ON** <*и м я т а бли цы* > (<*и м я ст о лбца* >

[,<*и м я ст о лбца* >]… );

При этом таблица д олжна быть уже созд ана и сод ержать столбцы, имена которых указ аны в команд е соз д ания инд екса. Имя инд екса, оп ред еленное в команд е, д олжно быть уникальным в базе д анных. Б уд уч и од нажд ы созд анным, инд екс яв ляется нев ид имым д ля п ользов ателя, в се оп ерации с ним осущ еств ляет СУ Б Д.

**П ример.**

Если таблица EXAM\_MARKS ч асто исп ольз уется д ля п оиска оценки конкретного студ ента п о знач ению п оля STUDENT\_ID, то след ует созд ать инд екс п о этому п олю.

**CREATE** **INDEX** STUDENT\_ID\_1 **ON** EXAM\_MARKS (STUDENT\_ID);

Для уд аления инд екса (п ри этом обязательно требуется з нать его имя) исп ользуется команд а **DROP** **INDEX**, имеющ ая след ующ ий синтаксис **DROP** **INDEX** <*и м я и нд екса* >;

У д аление инд екса не изменяет сод ержимого п оля или п олей, инд екс которых уд аляется.

**4.3. И зм енение сущ ест вую щ ей табл иц ы**

Для мод ификации структуры и п араметров сущ еств ующ ей таблицы исп ользуется команд а **ALTER** **TABLE**. Синтаксис команд ы **ALTER** **TABLE** д ля д обав ления столбцов в таблицу имеет в ид

**ALTER** **TABLE** <*и м я т а бли цы* > **ADD** (<*и м я ст о лбца* > <*т и п д а нны х*>

<*р а зм ер* >);

По этой команд е д ля сущ еств ующ их в таблице строк д обав ляется нов ый столбец, в который заносится **NULL**-знач ение. Этот столбец станов ится п ослед ним в таблице. М ожно д обав лять несколько столбцов , в этом случ ае их оп ред еления в команд е **ALTER** **TABLE** разд еляются зап ятой.

Возможно изменение оп исания столбцов . Часто это св язано с изменением размеров столбцов , д обав лением или уд алением огранич ений, наклад ыв аемых на их знач ения. Синтаксис команд ы в этом случ ае имеет в ид

**ALTER** **TABLE** <*и м я т а бли цы* > **MODIFY** <*и м я ст о лбца* > <*т и п д а нны х*>

<*р а зм ер /т о чно ст ь*>;

След ует иметь в в ид у, ч то мод ификация характеристик столбца может осущ еств ляться не в любом случ ае, а с уч етом след ующ их огранич ений:

* изменение тип а д анных в озможно только, если столбец п уст;
* д ля незап олненного столбца можно изменять размер/точ ность. Для з ап олненного столбца размер/точ ность можно ув елич ить, но нельзя п ониз ить.
* огранич ение **NOT** **NULL** может быть установ лено, если ни од но з нач ение в столбце не сод ержит **NULL**. О п цию **NOT** **NULL** в сегд а можно отменить.
* разрешается изменять знач ения, установ ленные п о умолч анию.

**4.4. Удал ение табл иц ы**

Чтобы уд алить сущ еств ующ ую таблицу, необход имо п ред в арительно уд алить в се д анные из этой таблицы, то есть сд елать ее п устой. Таблица, имеющ ая строки, не может быть уд алена. Синтаксис команд ы, осущ еств ляющ ей уд аление п устой таблицы, имеет след ующ ий в ид .

**DROP** **TABLE** <*и м я т а бли цы* >;

**УП РАЖ НЕ НИ Я**

1. Н ап ишите команд у **CREATE** **TABLE** д ля соз д ания таблицы LECTURER
2. Н ап ишите команд у **CREATE** **TABLE** д ля соз д ания таблицы SUBJECT 96. Н ап ишите команд у **CREATE** **TABLE** д ля соз д ания таблицы UNIVERSITY.
3. Н ап ишите команд у **CREATE** **TABLE** д ля соз д ания таблицы EXAM\_MARKS.
4. Н ап ишите команд у **CREATE** **TABLE** д ля соз д ания таблицы SUBJ\_LECT.
5. Н ап ишите кома нд у, которая п озв олит быстро в ыбрать д анные о студ ентах п о курсам, на которых они уч атся.
6. Соз д айте инд екс, который п оз в олит д ля кажд ого студ ента быстро осущ еств ить п оиск оценок, сгруп п иров анных п о д атам.

**4.5. О граничения на множ ество допустим ы х значений данны х**

До сих п ор рассматрив алось только след ующ ие огранич ения – з нач ения, в в од имые в таблицу, д олжны иметь тип ы д анных и размеры, сов местимые с тип ами/размером д анных столбцов , в которые эти з нач ения в в од ятся (как оп ред елено в команд е **CREATE** **TABLE** или **ALTER** **TABLE**). О п исание

таблицы может быть д оп олнено более сложными огранич ениями, наклад ыв аемыми на знач ения, которые могут быть в став лены в столбец или груп п у столбцов . О гранич ения (**CONSTRAINTS**) яв ляются ч астью оп ред еления таблицы.

При созд ании (изменении) таблицы могут быть оп ред елены огранич ения на в в од имые знач ения. В этом случ ае SQL буд ет отв ергать любое из них п ри не соотв етств ии зад анным критериям. О гранич ения могут быть статич ескими, огранич ив ающ ими знач ения или д иап азон знач ений, в став ляемых в столбец (**CHECH**, **NOT** **NULL**). О ни могут иметь св язь со в семи знач ениями столбца, огранич ив ая нов ые строки з нач ениями, которые не сод ержатся в столбцах или их наборах (уникальные з нач ения, п ерв ич ные ключ и). О гранич ения могут также оп ред еляться св яз ью со знач ениями, наход ящ имися в д ругой таблице, д оп уская, нап ример, в став ку в столбец только тех знач ений, которые в *д а нны й м о м ент* сод ержатся также в д ругом столбце д ругой или этой же таблицы (в нешний ключ ). Эти огранич ения носят д инамич еский характер.

Сущ еств ует д в а основ ных тип а огранич ений – огранич ения на столбцы и огранич ения на таблицу. О гранич ения на столбцы (**COLUMN** **CONSTRAINTS**) п рименимы только к отд ельным столбцам, а огранич ения на таблицу (**TABLE** **CONSTRAINTS**) п рименимы к груп п ам, состоящ им из од ного или более столбцов . О гра нич ения на столбец д обав ляются в конце оп ред еления столбца п осле указ ания тип а д анных и п еред оконч анием оп исания столбца (зап ятой). О гранич ения на таблицу размещ аются в конце оп ред еления таблицы, п осле оп ред еления п ослед него столбца. К оманд а **CREATE** **TABLE** имеет след ующ ий синтаксис, расширенный в ключ ением огранич ений

**CREATE** **TABLE** <*и м я т а бли цы* >

(*<и м я ст о лбца > <т и п д а нных> <о гр а ни чени я* *на ст о лбец*>,

<*и м я ст о лбца* > <*т и п д а нны х*> <*о гр а ни чени я на ст о лбец*>,…

<*о гр а ни чени я на т а бли цу*> (<*и м я ст о лбца* >[,<*и м я ст о лбца* >… ])… );

Поля, зад анные в круглых скобках п осле оп исания огранич ений таблицы – это п оля, на которые эти огранич ения расп ространяются. О гранич ения на столбцы п рименяются к тем столбцам, з а которыми они оп исаны.

**4.5.1. О граничение NOT NULL**

Чтобы з ап ретить в озможность исп ользов ания в п оле **NULL-**знач ений, можно п ри созд ании таблицы команд ой **CREATE** **TABLE** указать д ля соотв етств ующ его столбца ключ ев ое слов о **NOT** **NULL**. Это огранич ение п рименимо только к столбцам таблицы. К ак уже гов орилось в ыше, **NULL** – это сп ециальный маркер, обознач ающ ий тот факт, ч то п оле п усто. Н о он п олез ен не в сегд а. Перв ич ные ключ и, нап ример, в п ринцип е не д олжны сод ержать **NULL-**знач ений (быть п устыми), п оскольку это нарушило бы требов ание уникальности п ерв ич ного ключ а (более строго – функциональную зав исимость атрибутов таблицы от п ерв ич ного ключ а). Во многих д ругих случ аях также необход имо, ч тобы п оля *о бяза т ельно* сод ержали оп ред еленные з нач ения. Если ключ ев ое слов о **NOT** **NULL** размещ ается неп осред ств енно п осле тип а д анных (в ключ ая раз мер) столбца, то любые п оп ытки остав ить з нач ение п оля п устым (в в ести в п оле **NULL**знач ение) буд ут отв ергнуты системой.

Н ап ример, д ля того, ч тобы в оп ред елении таблицы **STUDENT** зап ретить исп ользов ание **NULL-**з нач ений д ля столбцов STUDENT\_ID, SURNAME и NAME, можно з ап исать след ующ ее:

**CREATE** **TABLE** STUDENT

(STUDENT\_ID INTEGER **NOT** **NULL**,

SURNAME **CHAR** (25) **NOT** **NULL**,

NAME **CHAR** (10) **NOT** **NULL**,

STIPEND **INTEGER**,

KURS **INTEGER**,

CITY **CHAR** (15),

BIRTHDAY **DATE**,

UNIV\_ID **INTEGER**);

Важно п омнить, ч то, если д ля столбца указано **NOT** **NULL**, то п ри исп ользов ании команд ы **INSERT** обязательно д олжно быть указано конкретное знач ение, в в од имое в это п оле. При отсутств ии огранич ения **NOT** **NULL** в столбце знач ение может отсутств ов ать, если только не указано знач ение столбца п о умолч анию (**DEFAULT**). Если п ри созд ании таблицы огранич ение **NOT** **NULL** не было указано, то его можно указ ать п оз же, исп ользуя команд у **ALTER** **TABLE**. О д нако, д ля того, ч тобы д ля в нов ь в в од имого с п омощ ью команд ы **ALTER** **TABLE** столбца можно было зад ать огранич ение **NOT** **NULL**, таблица, в которую д обав ляется столбец, д олжна быть п устой.

**4.5.2. Уникал ь ност ь как ограничение на ст ол бец**

Иногд а требуется, ч тобы в се знач ения, в в ед енные в столбец, отлич ались д руг от д руга. Н ап ример, этого требуют п ерв ич ные ключ и. Если п ри созд ании таблицы д ля столбца указыв ается огранич ение **UNIQUE**, то база д анных отв ергает любую п оп ытку в в ести в это п оле какой-либо строки знач ение, уже сод ержащ ееся в том же п оле д ругой строки. Это огранич ение п рименимо только к тем п олям, которые были объяв лены **NOT** **NULL**. Можно п ред ложить след ующ ее оп ред еление таблицы STUDENT, исп ользующ ее огранич ение **UNIQUE**.

**CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **NOT** **NULL** **UNIQUE**, |
| SURNAME | **CHAR** (25) **NOT** **NULL**, |
| NAME | **CHAR** (10) **NOT** **NULL**, |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |

BIRTHDAY **DATE**,

UNIV\_ID **INTEGER**);

О бъяв ляя п оле STUDENT\_ID уникальным, можно быть ув еренным, ч то в таблице не п ояв ится зап исей д ля д в ух студ ентов с од инаков ыми ид ентификаторами. Столбцы, отлич ные от п ерв ич ного ключ а, д ля которых требуется п од д ержать уникальность знач ений, наз ыв аются в оз можными ключ ами или уникальными ключ ами (**CANDIDATE** **KEYS** или **UNIQUE**

**KEYS**).

**4.5.3. Уникал ь ност ь как ограничение т абл иц ы**

М ожно сд елать уникальными груп п у п олей, указа в **UNIQUE** в кач еств е огранич ений *т а бли цы* . При объед инении п олей в груп п у в ажен п оряд ок, в котором они указыв аются. О гранич ение на таблицу **UNIQUE** яв ляется п олез ным, если требуется п од д ержив ать уникальность груп п ы п олей. Н ап ример, если в нашей базе д анных не д оп ускается, ч тобы студ ент сд ав ал в од ин д ень больше од ного экз амена, то можно в таблице объяв ить уникальной комбинацию знач ений п олей STUDENT\_ID и EXAM\_DATE. Д ля этого след ует созд ать таблицу EXAM\_MARKS таким сп особом.

**CREATE** **TABLE** EXAM\_MARKS

|  |  |
| --- | --- |
| (EXAM\_ID | **INTEGER NOT** **NULL**, |
| STUDENT\_ID | **INTEGER NOT** **NULL**, |
| SUBJ\_ID | **INTEGER NOT** **NULL**, |
| MARK | **CHAR** (1), |
| EXAM\_DATE | **DATE NOT** **NULL**, |

**UNIQUE** (STUDENT\_ID, EXAM\_DATE));

О братите в нимание, ч то оба п оля в огранич ении таблицы **UNIQUE** в се ещ е исп ользуют огранич ение столбца – **NOT** **NULL**. Если бы исп ользов алось огранич ение столбца **UNIQUE** д ля п оля STUDENT\_ID, то такое огранич ение таблицы было бы необязательным.

Если з нач ения п оля STUDENT\_ID д олжно быть раз лич ным д ля кажд ой строки в таблице EXAM\_MARKS, это можно сд елать, объяв ив **UNIQUE** как огранич ение самого п оля STUDENT\_ID. В этом случ ае не буд ет и д в ух строк с ид ентич ной комбинацией знач ений п олей STUDENT\_ID, EXAM\_DATE.

След ов ательно, указание **UNIQUE** как огранич ение таблицы наиболее п олез но исп ользов ать в случ аях, когд а не требуется уникальность инд ив ид уальных п олей, как это имеет место на самом д еле в рассматрив аемом п римере.

**4.5.4. П рисвоение им ен ограничения м**

О гранич ениям таблиц можно п рисв а ив ать уникальные имена. Преимущ еств о яв ного з ад ания имени огранич ения состоит в том, ч то в этом случ ае п ри в ыд ач е системой сообщ ения о нарушении установ ленного огранич ения буд ет указано его имя, ч то уп рощ ает обнаружение ошибок.

Для п рисв оения имени огранич ению исп ользуется несколько измененный синтаксис команд **CREATE** **TABLE** и **ALTER** **TABLE**.

Прив ед енный в ыше п ример зап роса изменяется след ующ им образом:

**CREATE** **TABLE** EXAM\_MARKS

|  |  |
| --- | --- |
| (EXAM\_ID | **INTEGER NOT** **NULL**, |
| STUDENT\_ID | **INTEGER NOT** **NULL**, |
| SUBJ\_ID | **INTEGER NOT** **NULL**, |
| MARK | **CHAR** (1), |
| EXAM\_DATE | **DATE NOT** **NULL**, |

**CONSTRAINT** STUD\_SUBJ\_CONSTR

**UNIQUE** (STUDENT\_ID, EXAM\_DATE);

В этом зап росе STUD\_SUBJ\_CONSTR – это имя, п рисв оенное указ анному огранич ению таблицы.

**4.5.5. О граничение первичны х кл ю чей**

*П ер ви чны е ключи т а бли цы* – это сп ециальные случ аи комбиниров ания огранич ений **UNIQUE** и **NOT NULL**. Перв ич ные ключ и имеют след ующ ие особенности:

* таблица может сод ержать только од ин п ерв ич ный ключ ;
* в нешние ключ и п о умолч анию ссыла ются на п ерв ич ный ключ таблицы;
* п ерв ич ный ключ яв ляется ид ентификатором строк таблицы (строки, од нако, могут ид ентифициров аться и д ругими сп особами).

У луч шенный в ариант соз д ания таблицы STUDENT1 с объяв ленным п ерв ич ным ключ ом имеет теп ерь след ующ ий в ид : **CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR** (25) **NOT** **NULL**, |
| NAME | **CHAR** (10) **NOT** **NULL**, |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |
| BIRTHDAY | **DATE**, |
| UNIV\_ID | **INTEGER**); |

**4.5.6. С ост авны е первичны е кл ю чи**

О гранич ение **PRIMARY** **KEY** может также быть п рименено д ля нескольких п олей, состав ляющ их уникальную комбинацию з нач ений – *со ст а вно й* п ерв ич ный ключ . Рассмотрим таблицу EXAM\_MARKS. О ч ев ид но, ч то ни к п олю ид ентификатора студ ента (STUDENT\_ID), ни к п олю ид ентификатора п ред мета обуч ения (EXAM\_ID) п о отд ельности нельзя п ред ъяв ить требов ание уникальности. О д нако, д ля того, ч тобы в таблице не могли п ояв иться разные зап иси д ля од инаков ых комбинаций знач ений п олей STUDENT\_ID и EXAM\_ID (конкретный студ ент на конкретном экзамене не может п олуч ить более од ной оценки), имеет смысл объяв ить уникальной комбинацию этих п олей. Для этого мы можем п рименить огранич ение таблицы **PRIMARY** **KEY**, объяв ив п ару EXAM\_ID и STUDENT\_ID п ерв ич ным ключ ом таблицы:

**CREATE** **TABLE** NEW\_EXAM\_MARKS

(STUDENT\_ID **INTEGER NOT** **NULL,**  SUBJ\_ID **INTEGER NOT** **NULL,**

MARK **INTEGER**,

DATA **DATE**,

**CONSTRAINT**  EX\_PR\_KEY **PRIMARY KEY** (EXAM\_ID, STUDENT\_ID));

**4.5.7. П роверка значений пол ей**

О гранич ение **CHECK** п озв оляет оп ред елять услов ие, которому д олжно уд ов летв орять в в од имое в п оле таблицы знач ение, п режд е ч ем оно буд ет п ринято. Любая п оп ытка обнов ить или заменить з нач ение п оля такими, д ля которых п ред икат, зад ав аемый огранич ением **CHECK**, имеет знач ение ло ж ь, буд ет отв ергаться.

Рассмотрим таблицу STUDENT. Знач ение столбца STIPEND в этой таблице STUDENT в ыражается д есятич ным ч ислом. Н аложим на знач ения этого столбца след ующ ее огранич ение – в елич ина размера стип енд ии д олжна быть меньше 200.

Соотв етств ующ ий зап рос имеет след ующ ий в ид .

**CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR** (25) **NOT** **NULL**, |
| NAME | **CHAR** (10) **NOT** **NULL**, |
| STIPEND | **INTEGER** **CHECK** (STIPEND < 200), |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |
| BIRTHDAY | **DATE**, |
| UNIV\_ID | **INTEGER**); |

**4.5.8. П роверка ограничиваю щ их усл овий с испол ь з ованием сост авны х пол ей**

О гранич ение **CHECK** можно исп ольз ов ать в кач еств е таблич ного огранич ения, то есть п ри необход имости в ключ ить более од ного п оля строки в огранич ив ающ ее услов ие.

Пред п оложим, ч то огранич ение на раз мер стип енд ии (меньше 200) д олжно расп ространяться только на студ ентов , жив ущ их в Воронеже. Это можно указать в зап росе со след ующ им таблич ным огранич ением **CHECK**:

**CREATE** **TABLE** **STUDENT**

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR**(25) **NOT** **NULL**, |
| NAME | **CHAR** (10) **NOT** **NULL**, |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR**(15), |
| BIRTHDAY | **DATE**, |
| UNIV\_ID | **INTEGER UNIQUE,** |

**CHECK**(STIPEND < 200 **AND** CITY **= ‘**Во р о н еж **’**)); или в несколько д ругой зап иси

**CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR**(25) **NOT** **NULL**, |
| NAME | **CHAR** (10) **NOT** **NULL**, |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR**(15), |
| BIRTHDAY | **DATE**, |

UNIV\_ID **INTEGER UNIQUE, CONSTRAINT** STUD\_CHECK **CHECK** (STIPEND < 200

**AND** CITY **= ‘**Во р о н еж **’**));

**4.5.9. Уст ановка значений по ум ол чанию**

В SQL имеется в оз можность п ри в став ке в таблицу строки, не указыв ая знач ений некоторого п оля, оп ред елять з нач ение этого п оля п о умолч анию. Н аиболее ч асто исп ольз уемым знач ением п о умолч анию яв ляется **NULL**. Это знач ение п ринимается п о умолч анию д ля любого столбца, д ля которого не было установ лено огранич ение **NOT** **NULL**.

Знач ение п оля п о умолч анию указыв ается в команд е **CREATE** **TABLE** тем же сп особом, ч то и огранич ение столбца, с п омощ ью ключ ев ого слов а **DEFAULT** <*зна чени е по ум о лча ни ю*>.

Строго гов оря, оп ция **DEFAULT** не имеет огранич ительного св ойств а, так как она не огранич ив ает з нач ения, в в од имые в п оле, а п росто конкретизирует знач ение п оля в случ ае, если оно *не бы ло за д а но* .

Пред п оложим, ч то основ ная масса студ ентов , информация о которых наход ится в таблице STUDENT, п рожив ает в Воронеже. Чтобы п ри з ад ании атрибутов не в в од ить д ля большинств а студ ентов наз в ание город а **‘**Во р о н еж **’**, можно установ ить его как знач ение п оля CITY п о умолч анию, оп ред елив таблицу STUDENT след ующ им образ ом:

**CREATE** **TABLE** STUDENT

( STUDENT\_ID **INTEGER** **PRIMARY** **KEY**, SURNAME **CHAR** (25) **NOT** **NULL**,

NAME **CHAR** (10) **NOT** **NULL**,

STIPEND **INTEGER** **CHECK** (STIPEND < 200),

KURS **INTEGER**,

CITY **CHAR** (15) **DEFAULT** **‘**Во р о н еж **’**,

BIRTHDAY **DATE**,

UNIV\_ID **INTEGER**);

Другая цель п рактич еского п рименения зад ания з нач ения п о умолч анию – это исп ользов ание его как альтернатив ы д ля **NULL**. К ак уже отмеч алось в ыше, п рисутств ие **NULL** в кач еств е в озможных з нач ений п оля сущ еств енно усложняет интерп ретацию оп ераций срав нения, в которых уч аств уют знач ения таких п олей, п оскольку **NULL** п ред став ляет собой п риз нак того, ч то фактич еское з нач ение п оля *неи звест но* или *нео пр ед еленно* . След ов ательно, строго гов оря, срав нение с ним любого конкретного знач ения в рамках д в узнач ной булев ой логики яв ляется не корректным, з а исключ ением сп ециальной оп ерации срав нения **IS** **NULL**, которая оп ред еляет, яв ляется ли сод ержимое п оля каким-либо знач ением или оно отсутств ует. Действ ительно, каким образом в рамках д в узнач ной логики отв етить на в оп рос истинно или ложно услов ие: CITY = **‘**Во р о н еж **’**, если текущ ее знач ение п оля CITY не из в естно (сод ержит **NULL**).

Во многих случ аях исп ользов ание в место **NULL** знач ения, п од став ляемого в п оле п о умолч анию, может сущ еств енно уп ростить исп ользов ание з нач ений п оля в п ред икатах.

Н ап ример, можно установ ить д ля столбца оп цию **NOT NULL**, а д ля неоп ред еленных з нач ений ч ислов ого тип а установ ить знач ение п о умолч анию “рав но нулю”, или д ля п олей тип а **CHAR** – п робел, исп ользов ание которых в оп ерациях срав нения не в ыз ыв ает никаких п роблем.

При исп ользов ании знач ений п о умолч анию в п ринцип е д оп устимо п рименять огранич ения **UNIQUE** или **PRIMARY** **KEY** в этом п оле. При этом, од нако, след ует иметь в в ид у отсутств ие в таком огранич ении п рактич еского смысла, п оскольку только од на строка в таблице сможет п ринять з нач ение, сов п ад ающ ее с этим знач ением п о умолч анию. Для такого п рименения зад ания п о умолч анию необход имо (д о в став ки д ругой строки с установ кой п о умолч анию) мод ифициров ать п ред ыд ущ ую строку, сод ержащ ую такое знач ение.

### УП РАЖ НЕ НИ Я

1. Соз д айте таблицу EXAM\_MARKS так, ч тобы не д оп ускался в в од в таблицу д в ух зап исей об оценках од ного студ ента п о конкретным экз амену и п ред мету обуч ения, а также, ч тобы не д оп ускалось п ров ед ение д в ух экзаменов п о любым п ред метам в од ин д ень.
2. Соз д айте таблицу п ред метов обуч ения SUBJECT так, ч тобы колич еств о отв од имых на п ред мет ч асов п о умолч анию было рав но 36, не д оп ускались зап иси с отсутств ующ им колич еств ом ч асов , п оле SUBJ\_ID яв лялось п ерв ич ным ключ ом таблицы, и з нач ения семестров (п оле SEMESTR) лежали в д иап азоне от 1-го д о 12-ти.
3. Соз д айте таблицу EXAM\_MARKS таким образом, ч тобы з нач ения п оля

EXAM\_ID были больше знач ений п оля SUBJ\_ID, а знач ения п оля SUBJ\_ID были больше знач ений п оля STUDENT\_ID; п усть также буд ут зап рещ ены знач ения **NULL** в любом из этих трех п олей.

**4.6. П оддерж ка ц ел остности данны х**

В таблицах рассматрив аемой базы д анных знач ения некоторых п олей св яз аны д руг с д ругом. Так п оле STUDENT\_ID в таблице STUDENT и п оле STUDENT\_ID в таблице EXAM\_MARKS св язаны тем, ч то оп исыв ают од ни и те же объекты, то есть сод ержат ид ентификаторы студ ентов , информация о которых хранится в баз е. Б олее того, з нач ения ид ентификаторов студ ентов , которые д оп устимы в таблице EXAM\_MARKS, д олжны в ыбираться только из сп иска з нач ений STUDENT\_ID, фактич ески п рисутств ующ их в таблице STUDENT, то есть п ринад лежащ их реально оп исанным в базе студ ентам. А налогич но, з нач ения п оля UNIV\_ID таблицы STUDENT д олжны соотв етств ов ать ид ентификаторам унив ерситетов UNIV\_ID, фактич ески п рисутств ующ им в таблице UNIVERSITY, а з нач ения п оля SUBJ\_ID таблицы EXAM\_MARKS д олжны соотв етств ов ать ид ентификаторам п ред метов обуч ения, фактич ески п рисутств ующ им в таблице SYBJECT.

О гранич ения, наклад ыв аемые указанным тип ом св яз и, наз ыв аются *о гр а ни чени ям и ссы ло чно й цело ст но ст и* . О ни состав ляют в ажную ч асть оп исания характеристик п ред метной области, обесп еч ения корректности д анных, хранящ ихся в таблицах. К оманд ы оп исания таблиц DML имеют сред ств а, п озв оляющ ие оп исыв ать огранич ения ссылоч ной целостности и обесп еч ив ать п од д ержание такой целостности п ри манип уляциях знач ениями п олей баз ы д анных.

* + 1. **В неш ние и родит ел ь ские кл ю чи**

К огд а кажд ое знач ение, п рисутств ующ ее в од ном п оле таблицы, п ред став лено в д ругом п оле д ругой или этой же таблицы, гов орят, ч то п ерв ое п оле ссылается на в торое. Это указыв ает на п рямую св яз ь межд у знач ениями д в ух п олей. Поле, которое ссылается на д ругое п оле, назыв ается ***в неш ним к лю ч ом*** , а п оле, на которое ссылается д ругое п оле, назыв ается ***р одит ельск им к лю ч ом*** . Так ч то п оле UNIV\_ID таблицы STUDENT – это в нешний ключ (оно ссылается на п оле д ругой таблицы), а п оле UNIV\_ID таблицы UNIVERSITY, *на ко т о р о е* ссылается этот в нешний ключ – это род ительский ключ .

Х отя в п рив ед енном п римере имена в нешнего и род ительского ключ ей сов п ад ают, они *не о бяза т ельно* д олжны быть од инаков ыми, хотя ч асто их сознательно з ад ают од инаков ыми, ч тобы соед инение было более нагляд ным.

* + 1. **С ост авны е внеш ние кл ю чи**

Н а п рактике в нешний ключ не обязательно может состоять только из од ного п оля. Под обно п ерв ич ному ключ у, в нешний ключ может состоять из любого ч исла п олей. Внешний ключ и род ительский ключ , на который он ссылается, конеч но же, д олжны быть оп ред елены на од инаков ом множеств е п олей (п о колич еств у п олей, тип ам п олей и п оряд ку след ов ания п олей). Внешние ключ и, состоящ ие из од ного п оля – п рименяемые в тип ов ых таблицах настоящ его из д ания, на иболее ч асты на п рактике. Чтобы сохранить п ростоту обсужд ения, буд ем гов орить о в нешнем ключ е, как об од иноч ном столбце, хотя в се, ч то буд ет излагаться о п оле, которое яв ляется в нешним ключ ом, сп рав ед лив о и д ля состав ных в нешних ключ ей, оп ред еленных на груп п е п олей.

* + 1. **С м ы сл внеш него и родит ел ь ского кл ю чей**

К огд а п оле яв ляется в нешним ключ ом, оно оп ред еленным образом св яз ано с таблицей, на которую этот ключ ссылается. К ажд ое знач ение в этом п оле (в нешнем ключ е) неп осред ств енно п рив яз ано к конкретному знач ению в д ругом п оле (род ительском ключ е). Знач ения род ительского ключ а д олжны быть уникальными, так он од нов ременно яв ляется ключ ом отношения. Знач ения в нешнего ключ а *не о бяза т ельно* д олжны быть уникальными, то есть в отношении может быть любое ч исло строк с од инаков ыми знач ениями атрибутов , яв ляющ ихся в нешним ключ ом. При этом строки, сод ержащ ие од инаков ые знач ения в нешнего ключ а д олжны *о бяза т ельно* ссылаться на конкретное, п рисутств ующ ее в д анный момент в таблице, з нач ение род ительского ключ а. К роме того, ни в од ной строке таблицы не д олжно быть знач ений в нешнего ключ а, д ля которых в текущ ий момент отсутств уют соотв етств ующ ие знач ения род ительского ключ а. Другими слов ами не д олжно быть так назыв аемых “в исяч их” ссылок в нешнего ключ а. Если указанные требов ания в ып олняются в конкретный момент сущ еств ов а ния баз ы д анных, то гов орят, ч то д анные наход ятся в *со гла сова нно м* состоянии, а сама база наход ится в состоянии *ссыло чно й цело ст но ст и* .

* + 1. **О граничение FOREIGN KEY (внеш него кл ю ча)**

Для решения в оп росов п од д ержания ссылоч ной целостности в SQL исп ользуется огранич ение **FOREIGN** **KEY**. Н азнач ение **FOREIGN** **KEY** – это огранич ение д оп устимых з нач ений п оля множеств ом знач ений род ительского ключ а, ссылка на который указ ыв ается п ри оп иса нии д анного огранич ения **FOREIGN** **KEY**.

Проблемы обесп еч ения ссылоч ной целостности в озникают как п ри в в од е знач ений п оля, яв ляющ егося в нешним ключ ом, так и п ри мод ификации/уд алении знач ений п оля, на которое ссылается этот ключ (род ительского ключ а). О д но из д ейств ий огранич ения **FOREIGN** **KEY** – это отклонение (блокиров ка) в в од а з нач ений в нешнего ключ а, отсутств ующ их в таблице с род ительским ключ ом. Также это огранич ение в озд ейств ует на в озможность изменять или уд а лять знач ения род ительского ключ а.

О гранич ение **FOREIGN** **KEY** исп ользуется в команд ах **CREATE** **TABLE** и **ALTER** **TABLE** п ри созд ании или мод ификации таблицы, которая сод ержит п оле, которое требуется объяв ить в нешним ключ ом. В кома нд е указыв ается имя род ительского ключ а, на который имеется ссылка в огранич ении **FOREIGN** **KEY**.

* + 1. **В неш ний кл ю ч как ограничение т абл иц ы**

Синтаксис огранич ения **FOREIGN** **KEY** имеет след ующ ий в ид .

**FOREIGN** **KEY** < *с п и с о к с т о л бцо в* >

**REFERENCES** < *ро ди т ел ь с к а я т а бл и ца* >

[< *ро ди т ел ь с к и й к л ю ч* >];

В этом п ред ложении *с п и с о к с т о л бцо в* – это сп исок из од ного или более столбцов таблицы, которые буд ут соз д аны или изменены команд ами **CREATE** **TABLE** или **ALTER** **TABLE** (д олжны быть отд елены д руг от д руга з ап ятыми).

Параметр *ро ди т ел ь с к а я т а бл и ца* – это имя таблицы, сод ержащ ей род ительский ключ . Это, в ч астности, может быть и таблицей, которая созд ается или из меняется текущ ей команд ой. Параметр *ро ди т ел ь с к и й к л ю ч* п ред став ляет собой сп исок столбцов род ительской таблицы, которые состав ляют собств енно род ительский ключ . О ба сп иска столбцов , оп ред еляющ их в нешний и род ительский ключ и, д олжны быть сов местимы, а именно:

* сод ержать од инаков ое ч исло столбцов .
* п ослед ов ательность (1-й, 2-й, 3-й и т.д .) столбцов сп иска в нешнего ключ а д олжны иметь тип ы д анных и раз меры, сов п ад ающ ие с соотв етств ующ ими (1-м, 2-м, 3-м и т.д .) столбцами сп иска род ительского ключ а.

Созд ад им таблицу STUDENT с п олем UNIV\_ID, оп ред еленным в кач еств е в нешнего ключ а, ссылающ егося на таблицуUNIVERSITY:

**CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| ( STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR** (25), |
| NAME | **CHAR** (10), |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |
| BIRTHDAY | **DATE**, |

UNIV\_ID **INTEGER REFERENCES**,

**CONSTRAINT** UNIV\_FOR\_KEY **FOREIGN** **KEY** (UNIV\_ID)

**REFERENCES** UNIVERSITY (UNIV\_ID));

При п рименении команд ы **ALTER** **TABLE** к какой-либо таблице д ля зад ания огранич ения **FOREIGN** **KEY**, знач ения в нешнего ключ а этой таблицы и род ительского ключ а соотв етств ующ ей таблицы д олжны наход иться в состоянии ссылоч ной целостности. В п ротив ном случ ае команд а буд ет отклонена.

Синтаксис команд ы **ALTER** **TABLE** в этом случ ае имеет след ующ ий в ид :

**ALTER** **TABLE** <*и м я т а бли цы* >

**ADD** **CONSTRAINT** < *и мя о гра ни чени я* >

**FOREIGN** **KEY** (< *с п и с о к с т о л бцо в внеш него к л ю ча* > )

**REFERENCES** < *и мя ро ди т ел ь с к о й т а бл и цы* >

[(< *с п и с о к с т о л бцо в ро ди т ел ь с к о го к л ю ча* >)];

Н ап ример, команд а

# ALTER TABLE STUDENT

**ADD** **CONSTRAINT** STUD\_UNIV\_FOR\_KEY

**FOREIGN** **KEY**  (UNIV\_ID)

**REFERENCES**UNIVERSITY (UNIV\_ID);

д обав ляет огранич ение в нешнего ключ а д ля таблицы STUDENT.

**4.6.6. В неш ний кл ю ч как ограничение ст ол бц ов**

О гранич ение в нешнего ключ а может указыв аться не д ля в сей таблицы, как это было п оказано в ыше, а неп осред ств енно на соотв етств ующ ий столбец таблицы. При таком в арианте, наз ыв аемом *ссы ло чны м о гр а ни чени ем ст о лбца* , ключ ев ое слов о **FOREIGN** **KEY** фактич ески не исп ольз уется. Просто исп ользуется ключ ев ое слов о **REFERENCES** и д алее указыв ается имя род ительского ключ а, п од обно след ующ ему п римеру.

# CREATE TABLE STUDENT

|  |  |
| --- | --- |
| ( STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR** (25), |
| NAME | **CHAR** (10), |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |
| BIRTHDAY | **DATE**, |

UNIV\_ID **INTEGER REFERENCES** UNIVERSITY(UNIV\_ID));

К оманд а оп ред еляет п оле STUDENT.UNIV\_ID как в нешний ключ , исп ользующ ий в кач еств е род ительского ключ а п оле

UNIVERSITY.UNIV\_ID, яв ляющ ееся ключ ом таблицы UNIVERSITY.

Эта форма экв ив алентна след ующ ему огранич ению таблицы STUDENT:

**FOREIGN** **KEY** (UNIV\_ID) **REGERENCES** UNIVERSITY (UNIV\_ID) или, в д ругой з ап иси,

**CONSTRAINT** UNIV\_FOR\_KEY **FOREIGN** **KEY** (UNIV\_ID)

**REFERENCES** UNIVERSITY (UNIV\_ID).

Если в род ительской таблице у род ительского ключ а указано огранич ение **PRIMARY** **KEY**, то п ри указ ании огранич ения **FOREIGN** **KEY**, наклад ыв аемого на таблицу или на столбцы, *м о ж но не ука зы ва т ь спи со к ст о лбцо в р о д и т ельско го ключа* . Естеств енно, в случ ае исп ольз ов ания ключ ей со многими п олями, п оряд ок столбцов в соотв етств ующ их в нешних и п ерв ич ных ключ ах д олжен сов п ад ать, и в любом случ ае, п ринцип сов местимости межд у д в умя ключ ами д олжен быть соблюд ен.

Н ап ример, если огранич ение **PRIMARY** **KEY** размещ ено в п оле UNIV\_ID таблицы UNIVERSITY

**CREATE** **TABLE** UNIVERSITY

( UNIV\_ID **INTEGER** **PRIMARY** **KEY**,

UNIV\_NAME **CHAR**(10),

RATING **INTEGER**,

CITY **CHAR**(15));

то в таблице STUDENT п оле UNIV\_ID можно исп ольз ов ать в кач еств е в нешнего ключ а, не указыв ая в ссылке имя род ительского ключ а:

**CREATE** **TABLE** STUDENT

|  |  |
| --- | --- |
| (STUDENT\_ID | **INTEGER** **PRIMARY** **KEY**, |
| SURNAME | **CHAR** (25), |
| NAME | **CHAR** (10), |
| STIPEND | **INTEGER**, |
| KURS | **INTEGER**, |
| CITY | **CHAR** (15), |
| BIRTHDAY | **DATE**, |

UNIV\_ID **INTEGER REFERENCES** UNIVERSITY);

Такая в озможность в страив алась в язык д ля обесп еч ения исп ользов ания

п ерв ич ных ключ ей в кач еств е род ительских.

**4.6.7. П оддерж ание ссы л очной ц ел ост ност и и ограничения значений родит ел ь ского кл ю ча**

Под д ержание ссылоч ной целостности требует в ып олнения некоторых огранич ений на з нач ения, которые могут быть зад аны в п олях, объяв ленных как в нешний ключ и род ительский ключ . Н абор знач ений род ительского ключ а д олжен быть таким, ч тобы гарантиров ать, ч то кажд ому знач ению в нешнего ключ а в род ительской таблице обязательно соотв етств ов ала од на и только од на строка, указанная соотв етств ующ им род ительским ключ ом. Это означ ает, ч то род ительский ключ д олжен быть *уни ка льны м* и не сод ержать п устых знач ений (**NULL**). След ов ательно, п ри объяв лении в нешнего ключ а необход имо убед иться, ч то в се п оля, которые исп ользуются как род ительские ключ и, имеют или огранич ение **PRIMARY** **KEY** или огранич ения **UNIQUE** и **NOT** **NULL**.

**4.6.8. И спол ь з ование первичного кл ю ча в качест ве уникал ь ного внеш него кл ю ча**

Ссылка в нешних ключ ей только на п ерв ич ные ключ и сч итается хорошим стилем п рограммиров ания SQL-зап росов . В этом случ ае исп ользуемые в нешние ключ и св языв аются не п росто с род ительскими ключ ами, на которые они ссылаются, а с од ной конкретной строкой род ительской таблицы, в которой буд ет найд ено соотв етств ующ ее знач ение род ительского ключ а. Сам п о себе род ительский ключ не обесп еч ив ает никакой информации, которая бы не была уже п ред став лена в о в нешнем ключ е. Внешний ключ – это не п росто св язь межд у д в умя ид ентич ными знач ениями столбцов д в ух таблиц, но это – св язь *м еж д у д вум я ст р ока м и д вух т а бли ц*.

Так как назнач ение п ерв ич ного ключ а состоит именно в том, ч тобы од нознач но ид ентифициров ать строку, то исп ользов ание ссылки на него в кач еств е в нешнего ключ а яв ляется более логич ным и более од нознач ным в ыбором д ля в нешнего ключ а. Внешний ключ , который не имеет никакой д ругой цели кроме св языв ания строк, нап оминает п ерв ич ный ключ , исп ользуемый исключ ительно д ля ид ентификации строк, и яв ляется хорошим сред ств ом сохранения нагляд ности и п ростоты структуры базы д анных.

**4.6.9. О граничения значений внеш него кл ю ча**

Внешний ключ может сод ержать только те з нач ения, которые фактич ески п ред став лены в род ительском ключ е, или яв ляются п устыми (**NULL**). Поп ытка в в ести д ругие знач ения в этот ключ д олжна быть отклонена, п оэтому объяв ление в нешнего ключ а, как **NOT** **NULL**, не яв ляется обязательным.

**4.6.10. Дей ст вие ограничений внеш него и родит ел ь ского кл ю чей при испол ь з овании ком анд м одиф икац ии**

К ак уже гов орилось, п ри исп ользов ании команд **INSERT** и **UPDATE** д ля мод ификации з нач ений столбца , объяв ленного как *внеш ни й ключ*, в нов ь в в од имые знач ения д олжны уже быть обяз ательно п ред став лены в фактич ески п рисутств ующ их знач ениях столбца, объяв ленного род ительским ключ ом. При этом можно п омещ ать в эти п оля п устые (**NULL**) знач ения, несмотря на то, ч то з нач ения **NULL** не д оп устимы в род ительских ключ ах. М ожно также уд алять (**DELETE**) любые строки с в нешними ключ ами из таблицы, в которой эти ключ и объяв лены.

При необход имости мод ификации з нач ений *р о д и т ельско го ключа* д ело обстоит инач е. Исп ользов ание команд ы **INSERT**, которая осущ еств ляет в в од нов ой з ап иси, не в ыз ыв ает никаких особенностей, п р и которых в озможно нарушение ссылоч ной целостности. О д нако кома нд а **UPDATE**, изменяющ ая знач ение род ительского ключ а и команд а **DELETE**, уд аляющ ая строку, сод ержащ ую такой ключ , сод ержат в озможность нарушения согласов анности знач ений род ительского и ссылающ ихся на него в нешних ключ ей. Н ап ример, может в оз никнуть так наз ыв аемая “*ви сяча я*” ссылка в нешнего ключ а на несущ еств ующ ее з нач ение род ительского ключ а, ч то сов ершенно не д оп устимо. Чтобы п ри п рименении команд **UPDATE** и **DELETE** к п олю, яв ляющ емуся род ительским ключ ом, не нарушалась целостность ссылки, в озможны след ующ ие в арианты д ейств ий.

* Л юбые из менения з нач ений род ительского ключ а *за пр еща ют ся* и п ри п оп ытке их сов ершения отв ергаются (огранич ение **NO** **ACTION** или **RESTRICT**). Эта сп ецификация д ейств ия п рименяется п о умолч а нию.
* Изменения з нач ений род ительского ключ а *р а зр еш а ют ся*, но п ри этом ав томатич ески осущ еств ляется *ко р р екци я* в сех знач ений в нешних ключ ей, ссылающ ихся на мод ифицируемое знач ение род ительского ключ а. Это назыв ается *ка ска д ны м и зм енени ем* (огранич ение **CASCADE**).
* Изменения з нач ений род ительского ключ а *р а зр еш а ют ся*, но п ри этом соотв етств ующ ие знач ения в нешнего ключ а ав томатич ески *уд а ляют ся*, то есть заменяются з нач ением **NULL** (огранич ение **SET** **NULL**).
* Изменения з нач ений род ительского ключ а *р а зр еш а ют ся*, но п ри этом соотв етств ующ ие з нач ения в нешнего ключ а а в томатич ески *за м еняют ся* з нач ением п о умолч а нию (огранич ение **SET** **DEFAULT**).

При оп исании в нешнего ключ а д олжно указыв аться, какой из п рив ед енных в ариантов д ейств ий след ует п рименять, п рич ем в общ ем случ ае это д олжно быть указ ано разд ельно д ля кажд ой из кома нд **UPDATE** и **DELETE**. В кач еств е п римера исп ользов ания огранич ений, на клад ыв аемых на оп ерации мод ификации род ительских ключ ей, можно п рив ести след ующ ий зап рос:

**CREATE** **TABLE** NEW\_EXAM\_MARKS

( STUDENT\_ID **INTEGER** **NOT** **NULL,**

SUBJ\_ID **INTEGER** **NOT** **NULL,**  MARK **INTEGER**,

DATA **DATE**,

**CONSTRAINT** EXAM\_PR\_KEY **PRIMARY KEY** (STUDENT\_ID, SUBJ\_ID), **CONSTRAINT**  SUBJ\_ID\_FOR\_KEY **FOREIGN KEY** (SUBJ\_ID)

**REFERENCES** SUBJECT,

**CONSTRAINT**  STUDENT\_ID\_FOR\_KEY **FOREIGN KEY** (STUDENT\_ID)  **REFERENCES** STUDENT **ON UPDATE CASCADE**

**ON DELETE** **NO** **ACTION**);

В этом п римере п ри п оп ытке из менения з нач ения п оля STUDENT\_ID таблицы STUDENT буд ет ав томатич ески обесп еч ив аться каскад ная корректиров ка этих з нач ений в таблице EXAM\_MARKS. То есть п ри изменении ид ентификатора студ ента STUDENT\_ID в таблице STUDENT сохранятся в се ссылки на его оценки. О д нако любая п оп ытка уд аления (**DELETE**) зап иси о студ енте из таблицы STUDENT буд ет отв ергаться, если в таблице EXAM\_MARKS сущ еств уют зап иси об оценках д анного студ ента.

### УП РАЖ НЕ НИ Я

1. Соз д айте таблицу с именем SUBJECT\_1, с теми же п олями, ч то в таблице SUBJECT (п ред мет обуч ения). Поле SUBJ\_ID яв ляется п ерв ич ным ключ ом.
2. Соз д айте таблицу с именем SUBJ\_LECT\_1 (уч ебные д исцип лины п реп од ав ателей), с п олями LECTURER\_ID (ид ентификатор п реп од ав ателя) и SUBJ\_ID (ид ентификатор п реп од ав аемой д исцип лины). Перв ич ным ключ ом (состав ным) таблицы яв ляется п ара атрибутов LECTURER\_ID и SUBJ\_ID, кроме того, п оле LECTURER\_ID яв ляется в нешним ключ ом, ссылающ имся на таблицу LECTURER\_1, ана логич ную таблице LECTURER (п реп од ав атель), а п оле SUBJ\_ID яв ляется в нешним ключ ом, ссылающ имся на таблицу SUBJECT\_1, аналогич ную таблице SUBJECT.
3. Соз д айте таблицу с именем SUBJ\_LECT\_1 как в п ред ыд ущ ем зад ании, но д обав ьте д ля в сех ее в нешних ключ ей режим обесп еч ения ссылоч ной целостности, зап рещ ающ ий обнов ление и уд аление соотв етств ующ их род ительских ключ ей.
4. Соз д айте таблицу с именем LECTURER\_1, с теми же п олями, ч то в таблице LECTURER. Перв ич ным ключ ом таблицы яв ляется атрибут LECTURER\_ID, кроме того, п оле UNIV\_ID яв ляется в нешним ключ ом, ссылающ имся на таблицу UNIVERSITY\_1 (а налог UNIVERSITY). Для этого п оля установ ите каскад ные режимы обесп еч ения целостности д ля команд **UPDATE** и **DELETE**.
5. Соз д айте таблицу с именем UNIVERSITY\_1, с теми же п олями, ч то в таблице UNIVERSITY (унив ерситеты). Поле UNIV\_ID яв ляется п ерв ич ным ключ ом.
6. Созд айте таблицу с именем EXAM\_MARKS\_1. О на д олжна сод ержать такие же п оля, ч то и таблица EXAM\_MARKS (экзаменационные оценки).

К омбинация п олей EXAM\_ID, STUDENT\_ID и SUBJ\_ID яв ляется п ерв ич ным ключ ом. К роме того, п оля STUDENT\_ID и SUBJ\_ID яв ляются в нешним ключ ами, ссылающ имися соотв етств енно на таблицы STUDENT\_1 и SUBJECT\_1. Д ля этих п олей установ ите режим каскад ного обесп еч ения ссылоч ной целостности п ри оп ерации обнов ления соотв етств ующ их п ерв ич ных ключ ей, и режим блокиров ки п ри п оп ытке уд аления род ительского ключ а п ри налич ии ссылки на него.

1. Соз д айте таблицу с именем STUDENT\_1. О на д олжна сод ержать такие же п оля, ч то и таблица STUDENT и нов ое п оле SENIOR\_STUDENT (староста), з нач ением которого д олжен быть ид ентификатор студ ента, яв ляющ егося старостой груп п ы, в которой уч ится д анный студ ент. У кажите необход имые д ля этого огранич ения ссылоч ной целостности.
2. Соз д айте таблицу STUDENT\_2 а налогич ную таблице STUDENT, в которой п оле UNIV\_ID (ид ентификатор унив ерситета) яв ляется в нешним ключ ом, ссылающ имся на таблицу UNIVERSITY\_1, и таким образ ом, ч тобы п ри уд а лении из таблицы UNIVERSITY\_1 строки с информацией о каком-либо унив ерситете в соотв етств ующ их зап исях таблицы STUDENT\_2 п оле UNIV\_ID оч ищ алось (замещ алось на **NULL**).
3. С п омощ ью команд ы **CREATE** **TABLE** соз д айте зап росы д ля формиров ания таблиц уч ебной базы д анных, п ред став ленной в раз д еле 1.7, с указанием п ерв ич ных ключ ей, но без указ ания огранич ений в нешних ключ ей. Затем с п омощ ью команд ы **ALTER** **TABLE** укажите д ля сформиров анных таблиц в се огранич ения, в том ч исле и огранич ения ссылоч ной целостности.

1. Пр едста влен и я **(VIEW)**

**5.1. П редставл ения – им енованны е запросы**

До сих п ор реч ь шла о таблицах, обыч но назыв аемых базов ыми таблицами. Это – таблицы, которые сод ержат д анные. О д нако имеется и д ругой в ид таблиц, назыв аемый **VIEW** или **ПРЕДСТАВЛЕНИЯ**. Таблицы– п ред став ления не сод ержат никаких собств енных д анных. Фактич ески ***пр едст а в ление*** *– эт о* ***им енов а нна я*** *т а бли ца , сод ер ж и м о е ко т ор ой являет ся р езульт а т о м за пр о са* , *за д а нно го пр и о пи са ни и пр ед ст а влени я*. Прич ем д анный зап рос в ып олняется в сякий раз, когд а таблица-п ред став ление станов ится объектом команд ы SQL. Выв од зап роса п ри этом в кажд ый момент станов ится сод ержанием п ред став ления. Пред став ления п оз в оляют:

* огранич ив ать ч исло столбцов , из которых п ользов атель в ыбирает или в которые в в од ит д а нные;
* огранич ив ать ч исло строк, из которых п ользов атель в ыбирает или в которые в в од ит д а нные;
* в ыв од ить д оп олнительные столбцы, п реобразов анные из д ругих столбцов базов ой таблицы;
* в ыв од ить груп п ы строк таблицы.

Б лагод аря этому п ред став ления д ают в озможность гибкой настройки в ыв од имой из таблиц информации в соотв етств ии с требов аниями конкретных п ользов ателей, п озв оляют обесп еч ив ать защ иту информации на уров не строк и столбцов , уп рощ ают формиров а ние сложных отч етов и в ыход ных форм.

Пред став ление оп ред еляется с п омощ ью кома нд ы **CREATE** **VIEW** (**СОЗДАТЬ** **ПРЕДСТАВЛЕНИЕ**). Н ап ример:

**CREATE** **VIEW** MOSC\_STUD **AS**

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

Данные из базов ой таблицы, п ред ъяв ляемые п ользов ателю в п ред став лении, за в исят от услов ия (п ред иката), оп исанного в **SELECT**зап росе п ри оп ред елении п ред став ления.

В созд анную в результате п рив ед енного в ыше зап роса таблицуп ред став ление MOSC\_STUD п еред аются д анные из базов ой таблицы STUDENT, но не в се, а только зап иси о студ ентах, д ля которых знач ение п оля CITY рав но ‘М о сква ’. К таблице MOSC\_STUD можно теп ерь обращ аться с п омощ ью зап росов так же, как и к любой д ругой таблице базы д анных.

Н а п ример, з ап рос д ля п росмотра п ред став ления MOSC\_STUD имеет в ид :

**SELECT** \*

**FROM** MOSC\_STUD;

**5.2. П редставл ения табл иц**

Раз лич ают п ред став ления таблиц и п ред став ления столбцов .

В п ростейшем п ред став лении таблиц в ыбираются *все* строки и столбцы базов ой таблицы.

**CREATE** **VIEW** NEW\_STUD\_TAB **AS**

**SELECT** \*

**FROM** STUDENT;

Такое п ред став ление, п о сути, экв ив алентно п рименению синонима, но яв ляется менее эффектив ным, п оэтому п рименяется ред ко.

**5.3. П редставл ения стол бц ов**

В п ростейшем в ид е п ред став ление столбцов в ыбирает в се строки и столбцы, п од обно п ред став лению таблиц; кроме того, в кач еств е имен столбцов п рименяются п сев д онимы. Н ап ример:

**CREATE** **VIEW** NEW\_STUDENT

(NEW\_STUDENT\_ID, NEW\_SURNAME, NEW\_NAME, NEW\_STIPEND,

NEW\_KURS, NEW\_CITY, NEW\_BIRTHDAY, NEW\_UNIV\_ID) **AS** **SELECT** STUDENT\_ID, SURNAME, NAME, STIPEND,

KURS, CITY, BIRTHDAY, UNIV\_ID

**FROM** STUDENT;

Пред став ление столбцов яв ляется п ростым сп особом организации общ ей таблицы д ля груп п ы п ользов ателей или п риклад ных зад ач , которые исп ользуют собств енные имена п олей и таблицы.

**5.4. Модиф иц ирование представл ений**

Данные, п ред ъяв ляемые п ользов ателю ч ерез п ред став ление, могут изменяться с п омощ ью команд мод ификации DML, но п ри этом фактич еская мод ификация д анных буд ет осущ еств ляться не в самой в иртуальной таблицеп ред став лении, а буд ет п ерена п рав лена к соотв етств ующ ей баз ов ой таблице.

Н ап ример, з ап рос на обнов ление п ред став ления NEW\_STUDENT

**UPDATE** NEW\_STUDENT

**SET** CITY = ‘М о сква ’

**WHERE** STUDENT\_ID = 1004; экв ив алентен в ып олнению команд ы **UPDATE** над базов ой таблицей STUDENT. След ует, од нако, обратить в нимание на то, ч то в общ ем случ ае, из-з а того, ч то обыч но в п ред став лении отображаются д анные из баз ов ой таблицы в *пр ео бр а зо ва нно м* или *усеченно м* в ид е, п рименение команд мод ификации к таблицам-п ред став лениям имеет некоторые особенности, рассматрив аемые ниже.

**5.5. Маскирую щ ие представл ения**

**5.5.1. П редст авл ения , маскирую щ ие ст ол бц ы**

Данный в ид п ред став лений огранич ив ает ч исло столбцов базов ой таблицы, к которым в озможен д оступ . Н ап ример, п ред став ление

**CREATE** **VIEW** STUD **AS**

**SELECT** STUDENT\_ID, SURNAME, CITY

**FROM** STUDENT;

д ает д оступ п ользов ателю к п олям STUDENT\_ID, SURNAME, CITY базов ой таблицы STUDENT, п олностью скрыв ая от него как сод ержимое, так и сам факт налич ия в базов ой таблице п олей NAME, STIPEND, KURS, BIRTHDAY и UNIV\_ID.

**5.5.2. О перац ии м одиф икац ии в предст авл ения х, маскирую щ их ст ол бц ы**

Пред став ления, как уже отмеч алось в ыше, могут изменяться с п омощ ью команд мод ификации DML, но п ри этом мод ификация д анных буд ет осущ еств ляться не в самой таблице-п ред став лении, а в соотв етств ующ ей базов ой таблице. В св яз и с этим, с п ред став лениями, маскирующ ими столбцы, функции в став ки и уд аления работают несколько инач е, ч ем с обыч ными таблицами. О п ератор **INSERT**, п римененный к п ред став лению, фактич ески осущ еств ляет в став ку строки в соотв етств ующ ую базов ую таблицу, п рич ем *во все* столбцы этой таблицы незав исимо от того, в ид ны они п ольз ов ателю ч ерез п ред став ление или скрыты от него. В св язи с этим, в столбцах, не в ключ енных в п ред став ление, устана в лив ается **NULL**значение или з нач ение п о умолч а нию. Если не в ключ енный в п ред став ление столбец имеет оп цию **NOT** **NULL**, то генерируется сообщ ение об ошибке.

Л юбое п рименение оп ератора **DELETE** уд аляет строки баз ов ой таблицы незав исимо от их з нач ений.

**5.5.3. П редст авл ения , маскирую щ ие ст роки**

Пред став ления могут также огранич ив ать д оступ к строкам. О хв атыв аемые п ред став лением строки базов ой таблицы зад аются услов ием (п ред икатом) в конструкции **WHERE** п ри оп исании п ред став ления. Доступ ч ерез п ред став ление в озможен только к строкам, уд ов летв оряющ им услов ию.

Н ап ример, п ред став ление

**CREATE** **VIEW** MOSC\_STUD **AS**

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

п оказ ыв ает п ольз ов ателю только те строки таблицы STUDENT, д ля которых знач ение п оля CITY рав но ‘М о сква ’.

**5.5.4. О перац ии м одиф икац ии в предст авл ения х, маскирую щ их ст роки**

К ажд ая в ключ енная в п ред став ление строка д оступ на д ля в ыв од а, обнов ления и уд аления. Любая д оп устимая д ля основ ной таблицы строка в став ляется в базов ую таблицу незав исимо от ее в ключ ения в п ред став ление. При этом может в озникнуть п роблема, состоящ ая в том, ч то з нач ения, в в ед енные п ользов ателем в базов ую таблицу ч ерез п ред став ление, знач ений, буд ут отсутств ов ать в п ред став лении, остав аясь п ри этом в баз ов ой таблице.

Рассмотрим такой случ ай:

**CREATE** **VIEW** HIGH\_RATING **AS**

**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING = 300;

Это п ред став ление яв ляется обнов ляемым. О но п росто огранич ив ает д оступ п ользов ателя к оп ред еленным столбцам и строкам в таблице UNIVERSITY. Пред п оложим, необход имо в став ить с п омощ ью команд ы **INSERT** след ующ ую строку:

**INSERT** **INTO** HIGH\_RATING

**VALUES** (180, ‘Н о вы й ун и вер си тет’, 200, ‘Во р о н еж ’);

К оманд а **INSERT** д оп устима в этом п ред став лении. С п омощ ью п ред став ления HIGH\_RATING строка буд ет в став лена в базов ую таблицу UNIVERSITY. О д нако, п осле п ояв ления этой строки в базов ой таблице, *и з са м о го пр ед ст а влени я о на и счезнет* , п оскольку знач ение п оля RATING не рав но 300, и, след ов ательно, эта строка не уд ов летв оряет услов ию п ред ложения **WHERE** д ля отбора строк в п ред став ление. Для п ользов ателя такое исч ез нов ение только ч то в в ед енной строки яв ляется неожид анным. Действ ительно, не п онятно, п оч ему п осле в в од а строки в таблицу ее нельзя ув ид еть и, нап ример, тут же уд алить. Тем более, ч то п ользов атель в ообщ е может не з нать – работает он в д анный момент с базов ой таблицей или с таблицей-п ред став лением.

А налогич ная ситуация в озникнет, если в какой-либо сущ еств ующ ей

зап иси п ред став ления HIGH\_RATING изменить з нач ение п оля RATING на знач ение, отлич ное от 300.

Под обные п роблемы можно устранить п утем в ключ ения в оп ред еление п ред став ления оп ции **WITH** **CHECK** **OPTION**. Эта оп ция расп ространяет услов ие **WHERE** д ля зап роса на оп ерации обнов ления и в став ки в оп исание п ред став ления. Н ап ример:

**CREATE** **VIEW** HIGH\_RATUNG **AS**

**SELECT** \*

**FROM** UNIVERSITY

**WHERE** RATING = 300

**WITH** **CHECK** **OPTION**;

В этом случ ае в ышеуп омянутые оп ерации в став ки строки или коррекции п оля RATING буд ет отклонены.

О п ция **WITH** **CHECK** **OPTION** п омещ ается в оп ред еление п ред став ления, а не в команд у **DML**, так ч то *все* команд ы мод ифика ции в п ред став лении буд ут п ров еряться. Рекоменд уется исп ользов ать эту оп цию в о в сех случ аях, когд а нет п рич ины разрешать п ред став лению п омещ ать в таблицу з нач ения, которые в нем самом не могут быть в ид ны.

**5.5.5. О перац ии м одиф икац ии в предст авл ения х, маскирую щ их ст роки и ст ол бц ы**

Рассмотренная в ыше п роблема в озникает и п ри в став ке строк в п ред став ление с п ред икатом, исп ользующ им п оля баз ов ой таблицы, не

п рисутств ующ ие в самом п ред став лении. Н ап ример, рассмотрим п ред став ление

**CREATE** **VIEW** MOSC\_STUD **AS**

**SELECT** STUDENT\_ID, SURNAME, STIPEND

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

Вид но, ч то в д анное п ред став ление не в ключ ено п оле CITY таблицы STUDENT.

Что буд ет п роисход ить п ри п оп ытках в став ки строки в это

п ред став ление? Так как мы не можем указать з нач ение CITY в п ред став лении как знач ение п о умолч анию (в в ид у отсутств ия в нем этого п оля), то этим знач ением буд ет **NULL**, и оно буд ет в в ед ено в п оле CITY базов ой таблицы STUDENT (сч итаем, ч то д ля этого п оля оп ция **NOT** **NULL** не исп ользуется). Так как в этом случ ае з нач ение п оля CITY баз ов ой таблицы STUDENT не буд ет рав няться знач ению ‘М о сква ’, в став ляемая строка буд ет исключ ена из самого п ред став ления и, п оэтому, не буд ет в ид на п ольз ов ателю. Прич ем так буд ет п роисход ить д ля *любо й* в став ляемой в п ред став ление MOSC\_STUD строки. Другими слов ами, п ользов атель в ообщ е не сможет в ид еть строки, в в од имые им в это п ред став ление. Данная п роблема не решается и в случ ае, если в оп ред еление п ред став ления буд ет д обав лена оп ция **WITH** **CHECK** **OPTION**

**CREATE** **VIEW** MOSC\_STUD **AS**

**SELECT** STUDENT\_ID, SURNAME, STIPEND

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’

**WITH** **CHECK** **OPTION**;

Таким образ ом, в оп ред еленном указанными сп особами п ред став лении, можно мод ифициров ать знач ения п олей или уд алять строки, но нельзя *вст а влят ь* строки. Исход я из этого, рекоменд уется д аже в тех случ аях, когд а этого не требуется п о соображениям п олез ности (и д аже безоп асности) информации, п ри оп ред елении п ред став ления в ключ ать в него *все* п оля, на которые имеется ссылка в п ред икате. Если эти п оля не д олжны отображаться в в ыв од е таблицы, в сегд а можно исключ ить их уже в з ап росе к п ред став лению. Другими слов ами, можно было бы оп ред елить п ред став ление MOSC\_STUD п од обно след ующ ему:

**CREATE** **VIEW** MOSC\_STUD **AS**

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’

**WITH** **CHECK** **OPTION**;

Эта команд а зап олнит в п ред став лении п оле CITY од инаков ыми знач ениями, которые можно п росто исключ ить из в ыв од а с п омощ ью д ругого зап роса уже к этому сформиров анному п ред став лению, указ ав в зап росе только п оля, необход имые д ля в ыв од а.

**SELECT** STUDENT\_ID, SURNAME, STIPEND  **FROM** MOSC\_STUD;

**5.6. Агрегированны е предст авл ения**

Созд ание п ред став лений с исп ользов анием агрегиров анных функций и п ред ложения **GROUP** **BY** яв ляется уд обным инструментом д ля неп рерыв ной обработки и интерп ретации изв лекаемой информации. Пред п оложим, необход имо след ить за колич еств ом студ ентов , сд ающ их экзамены, колич еств ом сд анных экзаменов , колич еств ом сд анных п ред метов , сред ним баллом п о кажд ому п ред мету. Для этого можно сформиров ать след ующ ее п ред став ление

**CREATE** **VIEW** TOTALDAY **AS**

**SELECT** EXAM\_DATE, **COUNT**(**DISTINCT** SUBJ\_ID) **AS** SUBJ\_CNT,  **COUNT**(STUDENT\_ID) **AS** STUD\_CNT,

**COUNT**(MARK**) AS** MARK\_CNT,

**AVG**(MARK) **AS** MARK\_AVG, **SUM**(MARK) **AS** MARK\_SUM  **FROM** EXAM\_MARKS

**GROUP** **BY** EXAM\_DATE;

Теп ерь требуемую информацию можно ув ид еть с п омощ ью п ростого зап роса к п ред став лению:

**SELECT** \* **FROM** TOTALDAY;

**5.7. П редставл ения , основанны е на нескол ь ких табл иц ах**

Пред став ления ч асто исп ользуются д ля объед инения нескольких таблиц

(баз ов ых и/или д ругих п ред став лений) в од ну большую в иртуальную таблицу. Такое решение имеет ряд п реимущ еств :

* п ред став ление, объед иняющ ее несколько таблиц, может исп ользов аться как п ромежуточ ный макет п ри формиров ании сложных отч етов , скрыв ающ ий д етали объед инения большого колич еств а исход ных таблиц.
* п ред в арительно объед иненные п оисков ые и баз ов ые таблицы обесп еч ив ают наилуч шие услов ия д ля транзакций, п оз в оляют исп ольз ов ать комп актные схемы код ов , устраняя необход имость нап исания д ля кажд ого отч ета д линных объед иняющ их п роцед ур.
* п озв оляет исп ользов ать п ри формиров ании отч етов более над ежный мод ульный п од ход .
* п ред в арительно объед иненные и п ров еренные п ред став ления уменьшают в ероятность ошибок, св язанных с неп олным в ып олнением услов ий объед инения.

М ожно, нап ример, созд ать п ред став ление, которое п оказыв ает имена и назв ания сд анных п ред метов д ля кажд ого студ ента:

**CREATE** **VIEW** STUD\_SUBJ **AS**

**SELECT** A.STUDENT\_ID, C.SUBJ\_ID, A.SURNAME, C.SUBJ\_NAME

**FROM** STUDENT A, EXAM\_MARKS B, SUBJECT C

**WHERE** A.STUDENT\_ID = B.STUDENT\_ID

**AND** B.SUBJ\_ID = C.SUBJ\_ID;

Теп ерь в се п ред меты студ ента или в сех студ ентов д ля кажд ого п ред мета можно в ыбрать с п омощ ью п ростого зап роса. Н ап ример, ч тобы ув ид еть в се п ред меты, сд анные студ ентом Ива н о вы м, п од ается зап рос:

**SELECT** SUBJ\_NAME

**FROM** STUD\_SUBJ

**WHERE** SURNAME = ‘Ива н о в’;

**5.8. П редставл ения и подзапросы**

При созд ании п ред став лений могут также исп ользов аться и п од зап росы, в ключ ая св язанные п од з ап росы. Пред п оложим, п ред усматрив ается п ремия д ля тех студ ентов , которые имеют самый в ысокий балл на любую з ад анную д ату. Получ ить такую информацию можно с п омощ ью п ред став ления:

**CREATE** **VIEW** ELITE\_STUD

**AS** **SELECT** B.EXAM\_DATE, A.STUDENT\_ID, A.SURNAME

**FROM** STUDENT A, EXAM\_MARKS B

**WHERE** A.STUDENT\_ID = B.STUDENT\_ID  **AND** B.MARK =

( **SELECT MAX (**MARK**)**

**FROM** EXAM\_MARKS C

**WHERE** C.EXAM\_DATE = B.EXAM\_DATE);

Если, с д ругой стороны, п ремия буд ет наз нач аться только студ енту, который имел самый в ысокий балл и не меньше 10-ти раз, то необход имо исп ользов ать д ругое п ред став ление, основ анное на п ерв ом:

**CREATE** **VIEW** BONUS

**AS SELECT DISTINCT** STUDENT\_ID, SURNAME  **FROM** ELITE\_STUD A  **WHERE** 10 < =

( **SELECT COUNT**(\*)

**FROM** ELITE\_STUD B

**WHERE** A.STUDENT\_ID = B.STUDENT\_ID);

Изв леч ение из этой таблицы з ап исей о студ ентах, которые буд ут п олуч ать п ремию, в ып олняется п ростым зап росом:

**SELECT** \* **FROM BONUS**;

**5.9. О граничения прим енения оператора SELECT дл я создания представл ений**

Имеются некоторые в ид ы зап росов , не д оп устимые в оп ред елениях п ред став лений. О д иноч ное п ред став ление д олжно основ ыв аться на од иноч ном з ап росе, п оэтому **UNION** и **UNION** **ALL** в п ред став лениях не разрешаются. Пред ложение **ORDER** **BY** также никогд а не исп ользуется в

оп ред елении п ред став лений. Пред став ление яв ляется реляционной таблицей-

отношением, п оэтому его строки п о оп ред елению яв ляются неуп оряд оч енными.

**5.10. Удал ение представл ений**

Синтаксис уд аления п ред став ления из базы д анных п од обен синтаксису уд аления базов ых таблиц:

**DROP** **VIEW** <*и мя п редст а вл ени я*>

**УП РАЖ НЕ НИ Я**

1. Соз д айте п ред став ление д ля п олуч ения св ед ений обо в сех студ ентах, имеющ их только отлич ные оценки.
2. Соз д айте п ред став ление д ля п олуч ения св ед ений о колич еств е студ ентов в кажд ом город е.
3. Соз д айте п ред став ление д ля п олуч ения св ед ений п о кажд ому студ енту: его ид ентификатор, фамилию, имя, сред ний и общ ий баллы.
4. Соз д айте п ред став ление д ля п олуч ения св ед ений о колич еств е экзаменов , которые сд ав ал кажд ый студ ент.

**5.11. И зм енение значений в представл ения х**

К ак уже гов орилось, исп ользов ание команд мод ификации яз ыка SQL – **INSERT (**ВСТАВИТЬ**)**, **UPDATE (**ЗАМЕНИТЬ**)**, и **DELETE (**УДАЛИТЬ**)** – п рименительно д ля п ред став лений имеет ряд особенностей. В д оп олнение к асп ектам, рассмотренным в ыше, след ует отметить, ч то не в се п ред став ления могут мод ифициров аться.

Если команд ы мод ификации могут в ып олняться в п ред став лении, то п ред став ление яв ляется обнов ляемым (мод ифицируемым); в п ротив ном случ ае оно п ред наз нач ено только д ля ч тения п ри зап росе. К аким образом можно оп ред елить, яв ляется ли п ред став ление мод ифицируемым? К ритерии обнов ляемости п ред став ления можно сформулиров ать след ующ им образом.

* Пред став ление строится на основ е од ной и только од ной баз ов ой таблицы.
* Пред став ление д олжно сод ержать п ерв ич ный ключ базов ой таблицы.
* Пред став ление не д олжно иметь никаких п олей, которые п ред став ляют собой агрегирующ ие функции.
* Пред став ление не д олжно сод ержать **DISTINCT** в св оем оп ред елении.
* Пред став ление не д олжно исп ользов ать **GROUP BY** или **HAVING** в св оем оп ред елении.
* Пред став ление не д олжно исп ольз ов ать п од з ап росы.
* Пред став ление *м о ж ет бы т ь* исп ользов ано в д ругом п ред став лении, но это п ред став ление д олжно быть также мод ифицируемыми.
* Пред став ление не д олжно исп ользов ать в кач еств е п олей в ыв од а константы или в ыражения з нач ений.

Суть этих огранич ений в том, ч то обнов ляемые п ред став ления фактич ески п од обны окна м в баз ов ых таблицах. О ни п оказыв ают информацию из баз ов ой таблицы, огранич ив ая оп ред еленные ее строки (исп ользов анием соотв етств ующ их п ред икатов ) или сп ециально именов анные столбцы (с исключ ениями). Н о п ри этом п ред став ления в ыв од ят з нач ения *без и х о бр а бо т ки* с исп ользов анием агрегирующ их функций и груп п иров ки. О ни также не срав нив ают строки таблиц д руг с д ругом (как это имеет место в объед инениях и п од зап росах, или п ри исп ользов ании **DISTINCT**).

Раз лич ия межд у мод ифицируемыми (обнов ляемыми) п ред став лениями и п ред став лениями “*т о лько д ля чт ени я*” не случ айны. О бнов ляемые п ред став ления в основ ном исп ольз уются аналогич но базов ым таблицам. Пользов атели могут д аже не знать, яв ляется ли зап рашив аемый ими объект базов ой таблицей или п ред став лением. Это п рев осход ный механиз м защ иты д ля скрытия ч астей таблицы, которые яв ляются конфид енциальными или не п ред назнач ены д анному п ользов ателю.

Н е мод ифицируемые п ред став ления, с д ругой стороны, п озв оляют более рационально п олуч ать и п ереформатиров ать д анные. С их п омощ ью формируются библиотеки сложных з ап росов , которые могут з атем исп ользов аться в зап росах д ля п олуч ения информации самостоятельно (нап ример, в объед инениях). Эти п ред став ления могут также иметь знач ение п ри решении зад ач защ иты и безоп асности д анных. Н ап ример, можно п ред остав ить некоторым п ользов ателям в озможность п олуч ения агрегатных д анных (таких, как усред ненное знач ение оценки студ ента), не п оказыв ая конкретных з нач ений оценок и, тем более, не п оз в оляя их мод ифициров ать.

**5.12. П рим еры обновл я ем ы х и не обновл я ем ы х предст авл ений**

**Пример 1**

**CREATE VIEW** DATEEXAM(EXAM\_DATE, QUANTITY)

**AS** **SELECT** EXAM\_DATE, **COUNT** (\*)

**FROM** EXAM\_MARKS

**GROUP** **BY** EXAM\_DATE;

Данное п ред став ление яв ляется ***не*** *о бно вляем ы м* из-з а п рисутств ия в нем агрегирующ ей функции и **GROUP** **BY**.

**Пример 2**

**CREATE** **VIEW** LCUSTT

**AS** **SELECT** \*

**FROM** UNIVERSITY  **WHERE** CITY = ‘М о сква ’;

Это – *о бно вляем о е* п ред став ление.

**Пример 3**

**CREATE** **VIEW** SSTUD (SURNAME1, NUMB, KUR)

**AS** **SELECT** SURNAME, STUDENT\_ID, KURS\*2

**FROM** STUDENT

**WHERE** CITY = ‘М о сква ’;

Это п ред став ление – ***не*** *м о д и ф и ци р уем о е* из-з а налич ия в ыражения “KURS\*2”.

**Пример 4**

**CREATE VIEW** STUD3

**AS SELECT \***

**FROM** STUDENT

**WHERE** STUDENT\_ID **IN**

( **SELECT** MARK

**FROM** EXAM\_MARKS

**WHERE** EXAM\_DATE = ‘10/02/1999’);

Пред став ление ***не*** *м о д и ф и ци р ует ся* из-за п рисутств ия в нем п од з ап роса.

В некоторых п рограммах это может быть п риемлемым.

**Пример 5**

**CREATE** **VIEW** SOMEMARK

**AS** **SELECT** STUDENT\_ID, SUBJ\_ID, MARK

**FROM** EXAM\_MARKS

**WHERE** EXAM\_DATE **IN** (‘10/02/1999’, ‘10/06/1999’); Это – *о бно вляем о е* п ред став ление.

**5.13. П редставл ения , базирую щ иеся на**

**других представл ения х**

О тносительно исп ользов ания п ред ложения **WITH** **CHECK** **OPTION** след ует отметить, ч то в станд арте SQL это п ред ложение не п ред усматрив ает каскад ного изменения, то есть оно п рименяется только в п ред став лениях, в которых оно оп ред елено, но не расп ространяется на д ругие п ред став ления, основ анные на этом п ред став лении. Н ап ример, в п ред ыд ущ ем п римере

**CREATE** **VIEW** HIGH\_RATING

**AS** **SELECT** UNIV\_ID, RATING

**FROM** UNIVERSITY

**WHERE** RATING >= 400

**WITH** **CHECK** **OPTION**;

п оп ытка в став ить или обнов ить знач ения п оля RATING, отлич ные от 400, буд ет отв ергнута, п оскольку п рисутств ует указание **WITH CHECK** **OPTION**. О д нако, если соз д ается в торое п ред став ление (с тем же сод ержанием), основ анное на п ерв ом:

**CREATE** **VIEW** MYRATING **AS**

**SELECT** \*

**FROM** HIGH\_RATING;

то в в од в п оле RATING с п омощ ью нижеп рив ед енного зап роса з нач ений, отлич ающ ихся от 400, уже не буд ет отв ергнуто как ошибоч ное. То есть след ующ ий зап рос

**UPDATE** MYRATING  **SET** RATING = 200

**WHERE** UNIV\_ID = 18;

не буд ет отв ергнут как не корректный, и, п осле его в ып олнения, строки с обнов ленными д анными исч ез нут из как из п ред став ления MYRATING, так и из п ред став ления HIGH\_RATING.

Пред ложение **WITH** **CHECK** **OPTION** п росто гарантирует, ч то любое обнов ление в п ред став лении осущ еств ляется в соотв етств ии со знач ениями, указ анными именно *д ля эт ого* п ред став ления. О бнов ление д ругих п ред став лений, базирующ ихся на п ерв ом текущ ем, п ри этом д оп устимым, если эти п ред став ления не з ащ ищ ены п ред ложениями **WITH** **CHECK** **OPTION**, зад анными именно д ля них. Пред ложения **WITH** **CHECK** **OPTION** п ров еряют п ред икаты только того п ред став ления, в котором они сод ержатся. При этом не яв ляется в ыход ом из п оложения и соз д ание п ред став ления MYRATING с п омощ ью зап роса

**CREATE** **VIEW** MYRATING **AS**

**SELECT** \*

**FROM** HIGH\_RATING

**WITH** **CHECK** **OPTION**;

**УП РАЖ НЕ НИ Я**

1. К акие из п ред став ленных ниже п ред став лений яв ляются обнов ляемыми?
   1. **CREATE** **VIEW** DAILYEXAM **AS**

**SELECT** **DISTINCT** STUDENT\_ID, SUBJ\_ID, MARK, EXAM\_DATE

**FROM** EXAM\_MARKS;

* 1. **CREATE VIEW** CUSTALS **AS**

**SELECT** SUBJECT.SUBJ\_ID, **SUM** (MARK) **AS** MARK1

**FROM** SUBJECT, EXAM\_MARKS

**WHERE** SUBJECT.SUBJ\_ID = EXAM\_MARKS.SUBJ\_ID

**GROUP** **BY** SUBJECT.SUBJ\_ID;

* 1. **CREATE VIEW** THIRDEXAM

**AS** **SELECT** \*

**FROM** DAILYEXAM

**WHERE** EXAM\_DATE = ‘10/02/1999’;

* 1. **CREATE VIEW** NULLCITIES

**AS** **SELECT** STUDENT\_ID, SURNAME, CITY

**FROM** STUDENT

**WHERE** CITY **IS** **NULL**

**OR** SURNAME **BETWEEN** ‘А’ **AND** ‘Д’;

1. Соз д айте п ред став ление таблицы STUDENT с именем STIP, в ключ ающ ее п оля STIPEND и STUDENT\_ID и п оз в оляющ ее в в од ить или изменять знач ение п оля STIPEND (стип енд ия), но только в п ред елах от 100 до 200.
2. О пр еделен и е пр а в до ступа по льзо ва телей к да н н ы м

**6.1. П ол ь зовател и и привил егии**

К ажд ый, кто имеет д оступ к базе д анных, наз ыв ается *по льзо ва т елем* . SQL исп ользуется обыч но в многоп ольз ов ательских сред ах, которые требуют разгранич ения п рав п ользов ателей с точ ки з рения д оступ а к д анным и п рав на в ып олнение с ними тех или иных манип уляций. Д ля этих целей в SQL реализов аны сред ств а, п оз в оляющ ие устанав лив ать и контролиров ать п рив илегии п ользов ателей базы д анных.

К ажд ый п ользов атель в сред е SQL имеет сп ециальное имя или ид ентификатор, с п омощ ью которого осущ еств ляется ид ентификация п ольз ов ателя д ля установ ки и оп ред еления его п рав с точ ки зрения д оступ а к д анным. К ажд ая п осланная к СУ Б Д команд а SQL-зап роса ассоциируется СУ Б Д с ид ентификатором д оступ а к д анным конкретного п ольз ов ателя.

Польз ов атель оп ред еляется с п омощ ью след ующ ей команд ы.

**CREATE USER** <*и м я\_по льзо ва т еля*> **IDENTIFIED BY** <*па р о ль*>

После в ып олнения этой команд ы п ользов атель станов ится из в естен базе д анных, но п ока не может в ып олнять никаких оп ераций. У д аление п ользов ателя п роизв од ится команд ой **DROP USER** <*и м я\_по льзова т еля*> .

Н азнач аемые п ольз ов ателю п рив илегии – это то, ч то оп ред еляет, может ли указанный п ользов атель в ып олнить д анную команд у над оп ред еленным объектом базы д анных или нет. Имеется несколько тип ов п рив илегий, соотв етств ующ их нескольким тип ам оп ераций. Прив илегии д аются и отменяются д в умя команд ами SQL, соотв етств енно:

**GRANT** – установ ка п рив илегий и **REVOKE** – отмена п рив илегий.

**6.2. С т андартны е привил егии**

Прив илегии, оп ред еленные станд артом SQL – это п рив илегии объекта. Это означ ает, ч то п ользов атель имеет п рив илегию (п рав о) на в ып олнение д анной команд ы только на оп ред еленном объекте в базе д анных. Прив илегии объекта св язаны од нов ременно и с п ольз ов ателями, и с таблицами базы д анных. То есть, п рив илегия д ается оп ред еленному п ользов ателю в указ анной таблице. Это может быть как базов ая таблица, так и п ред став ление.

Польз ов атель, созд ав ший таблицу (любого в ид а), яв ляется *вла д ельцем* этой таблицы. Это означ ает, ч то этот п ользов атель имеет *все* п рив илегии, относящ иеся к этой таблице, в том ч исле, он может п еред ав ать п рив илегии на работу с этой таблицей д ругим п ольз ов ателям.

Польз ов ателю могут быть наз нач ены след ующ ие п рив илегии:

* **SELECT** – п ользов атель может в ып олнять з ап росы к таблице;
* **INSERT** – п ользов атель может в ып олнять в таблицекоманд у **INSERT**;
* **UPDATE** – п ользов атель может в ып олнять в таблице команд у **UPDATE**.Эта п рив илегия может быть огранич ена д ля оп ред еленных столбцов таблицы;
* **DELETE** – п ользов атель может в ып олнять в таблице команд у **DELETE**;
* **REFERENCES** – п ользов атель может оп ред елить в нешний ключ (только д ля **ORACLE**), который исп ользует од ин или более столбцов этой таблицы, как род ительский ключ . Возможно огранич ение этой п рив илегии д ля оп ред еленных столбцов .

К роме того, могут быть нестанд артные п рив илегии объекта, такие, как:

* **INDEX** – п ользов атель имеет п рав о созд ав ать инд екс в таблице;
* **SYNONYM** – п ользов атель имеет п рав о созд ав ать синоним д ля объекта;
* **ALTER** – п ользов атель имеет п рав о в ып олнять команд у **ALTER** **TABLE** в таблице;
* **EXECUTE** – п озв оляет в ып олнять п роцед уру.

Н азнач ение п ольз ов ателям этих п рив илегий осущ еств ляется с п омощ ью команд ы **GRANT**.

**6.3. Ком анда GRANT**

Польз ов атель, яв ляющ ийся в лад ельцем таблицы STUDENT, может п еред ать д ругому п ользов ателю (п усть это буд ет п ользов атель с именем IVANOV) п рив илегию **SELECT** с п омощ ью след ующ ей кома нд ы.

**GRANT** **SELECT** **ON** STUDENT **TO** IVANOV;

Теп ерь п ользов атель с именем IVANOV может в ып олнять **SELECT**-з ап росы к таблице STUDENT. Б ез налич ия д ругих п рив илегий он может только *вы би р а т ь* з нач ения, но не может в ып олнять любые д ейств ия, которые бы в озд ейств ов али на з нач ения в таблице STUDENT, в ключ ая исп ользов ание таблицы STUDENT в кач еств е род ительской таблицы в нешнего ключ а. К огд а SQL п олуч ает команд у **GRANT**, п ров еряются п рив илегии п ользов ателя, д ав шего эту команд у, ч тобы оп ред елить д оп устимость команд ы **GRANT** д ля этого п ользов ателя. Польз ов атель IVANOV са мостоятельно не может зад ать эту команд у. О н также не может п ред остав ить п рав о **SELECT** д ругому п ольз ов ателю, так как таблица п ринад лежит не ему (ниже буд ет п оказано, как в лад елец таблицы может п еред ать д ругому п ользов ателю п рав о п ред остав ления п рив илегий).

К оманд а

**GRANT** **INSERT** **ON** EXAM\_MARKS **TO** IVANOV;

п ред остав ляет п ользов ателю IVANOV п рав о в в од ить в таблицу EXAM\_MARKS нов ые строки.

В команд е **GRANT** д оп устимо указ ыв ать ч ерез зап ятые сп исок п ред остав ляемых п рив илегий и сп исок п ользов ателей, которым они п ред остав ляются. Н ап ример:

**GRANT SELECT**, **INSERT** **ON** SUBJECT **TO** IVANOV, PETROV;

При этом в есь указанный сп исок п рив илегий п ред остав ляются в сем указанным п ользов ателям. В строгой ANSI-интерп ретации нев озможно п ред остав ить п рив илегии д ля нескольких таблиц од ной команд ой **GRANT**.

**6.4. И спол ь зование аргум ентов ALL и PUBLIC**

А ргумент **ALL** **PRIVILEGES** (в се п рив илегии) или п росто **ALL** исп ользуется в место имен п рив илегий в команд е **GRANT**, ч тобы п ред остав ить в се п рив илегии в таблице. Н а п ример, команд а **GRANT** **ALL** **PRIVILEGES** **ON** STUDENT **TO** IVANOV; или более коротко

**GRANT** **ALL** **ON** STUDENT **TO** IVANOV;

п еред ает п ольз ов ателю IVANOV ***в есь*** набор п рив илегий в таблице STUDENT.

А ргумент **PUBLIC** исп ольз уется д ля п еред ач и указанных в команд е п рив илегий *всем* остальным п ольз ов ателям. Н аиболее ч асто это п рименяется д ля п рив илегии **SELECT** в оп ред еленных базов ых таблицах или п ред став лениях, которые необход имо сд елать д оступ ными д ля любого п ольз ов ателя. Н ап ример, ч тобы п озв олить любому п ользов ателю п олуч ать информацию из таблицы EXAM\_MARKS, можно исп ользов ать команд у

**GRANT** **SELECT** **ON** EXAM\_MARKS **TO** **PUBLIC**;

Пред остав ление *всех* п рив илегий к таблице *всем* п ользов ателям обыч но яв ляется нежелательным. Все п рив илегии за исключ ением **SELECT** п озв оляют п ользов ателю из менять (или, в случ ае **REFERENCES**, огранич ив ать) сод ержание таблицы, п оэтому разрешение в сем п ользов ателям изменять сод ержание таблиц может в ыз в ать оп ред еленные п роблемы обесп еч ения безоп асности и защ иты д анных. Тем более, ч то п рив илегия **PUBLIC** не огранич ена в п еред ач е п рав только текущ им п ольз ов ателям. Л юбой нов ый п ольз ов атель, д обав ляемый к системе, ав томатич ески п олуч ает в этом случ ае п олный набор п рив илегий, назнач енный ранее в сем п ольз ов ателям. Поэтому д ля огранич ения д оступ а к таблице в сем и в сегд а луч ше в сего п ред остав ить п рив илегии, отлич ные от **SELECT**, только инд ив ид уальным п ользов ателям.

**6.5. О т м ена привил егий**

О тмена п рив илегии осущ еств ляется с п омощ ью команд ы **REVOKE**, которая имеет синтаксис, аналогич ный команд е **GRANT**.

Н ап ример, команд а

**REVOKE** **INSERT** **ON** STUDENT **FROM** PETROV; отменяет п рив илегию **INSERT** в таблице STUDENT д ля п ользов ателя PETROV. Воз можно исп ользов ание в команд е **REVOKE** сп исков п рив илегий и п ольз ов ателей. Н ап ример

**REVOKE** **INSERT**, **DELETE** **ON** STUDENT

**FROM** PETROV, SIDOROV;

След ует иметь в в ид у, ч то п рив илегии отменяются тем п ольз ов ателем, который их п ред остав ил, и, п ри этом отмена ав томатич ески расп ространяется на в сех п ользов ателей, п олуч ив ших от него эту п рив илегию.

**6.6. И спол ь зование представл ений дл я ф ил ь трац ии привил егий**

Действ ия п рив илегий можно сд елать более точ ными, исп ольз уя п ред став ления. При п еред ач е п рив илегии п ольз ов ателю в базов ой таблице, она ав томатич ески расп ространяется на в се строки, а п ри исп ольз ов ании в озможных исключ ений **UPDATE** и **REFERENCES**, и на в се столбцы таблицы. Созд ав ая п ред став ление, которое ссылается на баз ов ую таблицу, и затем, п еред ав ая п р ив илегию уже на это п ред став ление, можно огранич ить эти п рив илегии любыми в ыражениями в зап росе, сод ержащ емся в п ред став лении. Такой метод расширяет в озможности команд ы **GRANT**.

Для соз д ания п ред став лений п ользов атель д олжен облад ать п рив илегией **SELECT** в о в сех таблица х, на которые он ссылается в п ред став лении. Если п ред став ление мод ифицируемое, то любая из п рив илегий **INSERT**, **UPDATE** и **DELETE**, которая п ред остав лена п ольз ов ателю в базов ой таблице, буд ет ав томатич ески расп ространяться на п ред став ление. Если п рив илегии на обнов ление отсутств уют, то их нев озможно п олуч ить и в созд анных п ред став лениях, д аже если сами эти п ред став ления обнов ляемые. Так как в нешние ключ и не п рименяются в п ред став лениях, то и п рив илегия **REFERENCES** никогд а не исп ользуется п ри созд ании п ред став лений.

**6.6.1. О граничение привил егии SELECT дл я определ енны х ст ол бц ов**

Пред п оложим, необход имо обесп еч ить п ользов ателю PETROV в озможность д оступ а только к столбцам STUDENT\_ID и SURNAME таблицы STUDENT. Это можно сд елать, п оместив имена этих столбцов в п ред став ление

**CREATE** **VIEW** STUDENT\_VIEW **AS**

**SELECT** STUDENT\_ID, SURNAME

**FROM** STUDENT;

и п ред остав ить п ольз ов ателю PETROV п рив илегию **SELECT** в соз д анном п ред став лении, а не в самой таблице STUDENT:

**GRANT** **SELECT** **ON** STUDENT\_VIEW **TO** PETROV;

Для столбцов можно созд ать различ ные п рив илегии, од нако, след ует иметь в в ид у, ч то д ля команд ы **INSERT** это буд ет означ ать в став ку знач ений п о умолч анию, а д ля команд ы **DELETE** огранич ение столбца не буд ет в ообщ е иметь з нач ения.

**6.6.2. О граничение привил егий дл я определ енны х ст рок**

Пред став ления п озв оляют огранич ить (фильтров ать) п рив илегии д ля оп ред еленных строк таблицы. Д ля этого естеств енно исп ольз ов ать в п ред став лении п ред икат, который оп ред елит, какие строки в ключ ены в п ред став ление. Чтобы п ред остав ить п ользов ателю PETROV п рив илегию в ид а **UPDATE** в таблице UNIVERSITY д ля в сех зап исей о Москов ских унив ерситетах, можно созд ать след ующ ее п ред став ление:

**CREATE** **VIEW** MOSC\_UNIVERSITY **AS**

**SELECT** \* **FROM** UNIVERSITY

**WHERE** CITY = ‘М о сква ’  **WITH** **CHECK** **OPTION**; Затем можно п еред ать п рив илегию **UPDATE** в этой таблице п ольз ов ателю PETROV:

**GRANT** **UPDATE** **ON** MOSC\_UNIVERSITY **TO** PETROV;

В отлич ие от п рив илегии **UPDATE** д ля оп ред еленных столбцов , которая расп ространена на в се строки таблицы UNIVERSITY, д анная п рив илегия относится только к строкам, д ля которых знач ение п оля CITY рав но ‘М о сква ’. Пред ложение **WITH** **CHECK** **OPTION** п ред охраняет п ользов ателя PETROV от замены знач ения п оля CITY на любое знач ение, кроме знач ения ‘М о сква ’.

**6.6.3. П редост авл ение дост упа т ол ь ко к из вл еченны м данны м**

Другая в озможность состоит в том, ч тобы устана в лив ать п ользов ателям п рив илегии на д оступ к уже изв леч енным д анным, а не к знач ениям в таблице. Для этого уд обно исп ользов ать агрегирующ ие функции. Н ап ример, созд ад им п ред став ление, которое д ает информацию о колич еств е оценок, сред нем и общ ем ба ллах д ля студ ентов на кажд ый д ень:

**CREATE** **VIEW** DATETOTALS **AS**

**SELECT** EXAM\_DATE, **COUNT** (\*) **AS** KOL, **SUM** (MARK) **AS** SUMMA,

**AVG** (MARK) **AS** TOT

**FROM** EXAM\_MARKS

**GROUP** **BY** EXAM\_DATE;

Теп ерь можно п еред ать п ользов ателю PETROV п рив илегию **SELECT** в созд анном п ред став лении DATETOTALS с п омощ ью зап роса:

**GRANT** **SELECT** **ON** DATETOTALS **TO** PETROV;

**6.6.4. И спол ь з ование предст авл ений в качест ве ал ь т ернат ивы ограничения м**

Пред став ления с **WITH** **CHECK** **OPTION** могут исп ользов аться в кач еств е альтернатив ы огранич ениям. Н ап ример, необход имо уд остов ериться, ч то в се знач ения п оля CITY в таблице STUDENT рав ны наз в аниям конкретных городов . Для этого можно установ ить огранич ение **CHECK** неп осред ств енно на столбец CITY. О д нако п озже его изменение буд ет затруд нено. В кач еств е альтернатив ы можно созд ать п ред став ление, исключ ающ ее неп рав ильные знач ения CITY:

**CREATE VIEW** CURCITYES **AS**

**SELECT** \*

**FROM** STUDENT

**WHERE** CITY **IN** (‘М о сква ’, ‘Во р о н еж ’)

**WITH** **CHECK** **OPTION**;

Теп ерь, в место того, ч тобы п ред остав лять п ольз ов ателям п рив илегии обнов ления в таблице STUDENT, можно п ред остав ить соотв етств ующ ие п рив илегии в п ред став лении CURCITYES. Преимущ еств о такого п од ход а состоит в том, ч то п ри необход имости изменения можно уд алить это п ред став ление, созд ать нов ое, и п ред остав ить в этом нов ом п ред став лении п рив илегии п ользов ателям. Такая оп ерация в ып олняется п рощ е, ч ем изменение огранич ений в таблице. Н ед остатком этого метод а яв ляется то, ч то в лад елец таблицы STUDENT также д олжен исп ользов ать это нов ое п ред став ление, инач е его собств енные команд ы также не буд ут п риняты.

**6.7. Д ругие типы привил егий**

До сих п ор не рассмотрены в оп росы установ ки целого ряд а д ругих п рив илегий, а именно:

* К то имеет п рав о созд ав ать таблицы?
* К то имеет п рав о изменять, уд алять, или огранич ив ать таблицы?
* Должны ли п рав а созд ания базов ых таблиц отлич аться от п рав соз д ания п ред став лений?
* Должен ли сущ еств ов ать *супер по льзо ва т ель*, то есть п ользов атель, отв еч ающ ий за п од д ержание баз ы д анных и, след ов ательно, имеющ ий наибольшие, или п олные п рив илегии, которые не п ред остав ляются обыч ному п ользов ателю?

Прив илегии, которые не оп ред еляются в терминах сп ециальных объектов д анных, наз ыв аются *пр и ви леги ям и си ст ем ы* , или *пр а ва м и ба зы д а нны х*. Эти п рив илегии в ключ ают в себя п рав о соз д ав ать объекты д анных, отлич ающ иеся от баз ов ых таблиц (обыч но соз д ав аемых несколькими п ольз ов ателями) и п ред став лений (обыч но созд ав аемых большинств ом п ольз ов ателей). Прив илегии системы д ля созд ания п ред став лений д олжны д оп олнять, а не заменять п рив илегии объекта, которые станд арт требует от созд ателей п ред став лений (оп исаны ранее). К роме того, в любой системе в сегд а имеются некоторые тип ы суп ерп ользов ателей, то есть п ользов ателей, которые имеют большинств о или в се п рив илегии, и которые могут п еред ать св ой статус суп ерп ольз ов ателя кому-либо с п омощ ью п рив илегии или груп п ы п рив илегий. Такого род а п ольз ов ателем яв ляется так назыв аемый *а д м и ни ст р а т о р ба зы д а нны х*, или DBA (DataBase Administrator)**.**

**6.8. Т ипичны е привил егии систем ы**

При общ ем п од ход е имеется три базов ых п рив илегии системы:

* **CONNECT** (Подключить),
* **RESOURCE** (Ресурс) и
* **DBA** (Администратор Базы Данных).

Прив илегия **CONNECT** состоит из п рав а зарегистриров аться и п рав а созд ав ать п ред став ления и синонимы, если п еред аны п рив илегии объекта.

Прив илегия **RESOURCE** состоит из п рав а соз д ав ать базов ые таблицы.

Прив илегия **DBA** – это п рив илегия ад министратора баз ы д анных, то есть суп ерп ользов ателя, которому п ред остав ляются самые в ысокие п олномоч ия п ри работе с баз ой д анных. Эту п рив илегию может иметь од ин или более п ольз ов ателей с функциями ад министратора базы д анных. К оманд а **GRANT** (в измененной форме) может п рименяться как с п рив илегиями объекта, так и с системными п рив илегиями.

**6.9. С оздание и удал ение пол ь зоват ел ей**

В большинств е реализаций SQL нов ого п ользов ателя созд ает п ольз ов атель с п рив илегией **DBA**, то есть ад министратор баз ы д анных, который ав томатич ески п ред остав ляет нов ому п ольз ов ателю п рив илегию **CONNECT**. В этом случ ае обыч но д обав ляется п ред ложение **IDENTIFIED** **BY**, указ ыв ающ ее п ароль д ля этого п ользов ателя. Н ап ример, команд а

**GRANT** **CONNECT** **TO** PETROV **IDENTIFIED** **BY** ‘PETROVPASSWORD’; п рив ед ет к соз д анию п ользов ателя с именем PETROV, п ред остав ит ему п рав о регистриров аться в базе д анных, и наз нач ает ему п ароль “PETROVPASSWORD”. После этого, так как PETROV уже яв ляется зарегистриров анным п ольз ов ателем, он (или п ользов атель **DBA)** может исп ользов ать эту же команд у д ля изменения д анного п ароля “PETROVPASSWORD”.

К огд а п ольз ов атель A п ред остав ляет п рив илегию **CONNECT** д ругому п ольз ов ателю B, гов орят, ч то п ользов атель A “*со зд а ет* ” п ольз ов ателя B. При этом п ользов атель A обяз ательно д олжен иметь п рив илегию **DBA**. Если п ольз ов атель B буд ет созд ав ать базов ые таблицы (а не только п ред став ления), ему также д олжна быть п ред остав лена п рив илегия **RESOURCE**. Н о п ри этом в озникает д ругая п роблема. При п оп ытке уд аления п ольз ов ателем A п рив илегии **CONNECT** п ользов ателя B, который уже имеет созд анные им таблицы, эта команд а уд аления п рив илегии буд ет отклонена, п оскольку ее д ейств ие остав ит эти таблицы без в лад ельца, ч то не д оп ускается. Поэтому, п режд е ч ем уд алить п рив илегию **CONNECT** какомулибо п ольз ов ателю, снач ала необход имо уд алить из базы д анных в се созд анные этим п ользов ателем таблицы. Прив илегию **RESOURCE** уд алять отд ельно не требуется, д остаточ но уд алить **CONNECT**, ч тобы уд алить п ольз ов ателя.

**УП РАЖ НЕ НИ Я**

1. Перед айте п ольз ов ателю PETROV п рав о на изменение в базе д анных з нач ений оценок д ля з ап исей о студ ентах.
2. Перед айте п ользов ателю SIDOROV п рав о п еред ав ать д ругим п ользов ателям п рав а на осущ еств ление зап росов к таблице EXAM\_MARKS.
3. О тмените п рив илегию **INSERT** п о отношению к таблице STUDENT у п ользов ателя IVANOV и у в сех д ругих п ольз ов ателей, которым п рив илегия, в св ою оч еред ь, была п ред остав лена этим п ользов ателем IVANOV.
4. Перед айте п ольз ов ателю SIDOROV п рав о в ып олнять оп ерации в став ки или обнов ления д ля таблицы UNIVERSITY, но только д ля зап исей об унив ерситетах, знач ения рейтингов которых лежат в д иап аз оне от 300 до 400.
5. Разрешите п ользов ателю PETROV д елать за п росы к таблице EXAM\_MARKS, но зап ретите ему изменять в этой таблице з нач ения оценок студ ентам, имеющ им неуд ов летв орительные (=2) оценки.

**6.10. С оздание синонимов (SYNONYM)**

К ажд ый раз п ри ссылке к базов ой таблице или п ред став лению, не яв ляющ имися собств енностью п ользов ателя, требуется установ ить в кач еств е п рефикса к имени этой таблицы имя ее в лад ельца, п оскольку система не сможет оп ред елить местонахожд ение таблицы, так как у разных п ольз ов ателей могут оказаться таблицы с од инаков ыми именами. Исп ольз ов ание д линных имен с п рефиксами может оказаться неуд обным. Поэтому большинств о реализаций SQL п оз в оляют созд ав ать д ля таблиц синонимы (ч то не яв ляется станд артом ANSI). *Си но ни м* – это альтернатив ное имя таблицы. При созд ании синонимов п ольз ов атель станов ится его собств енником, п оэтому необход имость исп ольз ов ания п рефикса к имени таблицы д ля него отп ад ает. Пользов атель имеет п рав о созд ав ать синоним д ля таблицы, если он имеет, п о крайней мере, од ну п рив илегию в од ном или более столбцах этой таблицы.

С п омощ ью команд ы **CREATE** **SYNONYM** п ольз ов атель IVANOV может д ля таблицы с именем PETROV.STUDENT созд ать синоним с именем CLIENTS след ующ им образом.

**CREATE SYNONYM** CLIENTS **FOR** PETROV.STUDENT;

Теп ерь п ользов атель IVANOV может исп ользов ать таблицу с именем CLIENTS в команд е точ но так же, как имя PETROV.STUDENT.

К ак уже гов орилось, п рефикс п ользов ателя – это фактич ески ч асть имени любой таблицы. Всякий раз, когд а п ользов атель не указ ыв ает собств енное имя в месте с именем св оей таблицы, SQL п о умолч анию п од став ляет ид ентификатор п ользов ателя в кач еств е п рефикса имени таблицы. След ов ательно, д в а од инаков ых имени таблицы, но св язанные с различ ными в лад ельцами, станов ятся неид ентич ными и, след ов ательно, не п рив од ят к какой-либо п утанице в з ап росах. Та ким образом, д в а п ольз ов ателя могут созд ав ать д в е п олностью незав исимые таблицы с од инаков ыми именами, но это также оз нач ает, ч то од ин п ользов атель может созд ать п ред став ление, основ анное на имени, стоящ ем п осле имени таблицы, и исп ользуемым д ругим п ользов ателем. Это иногд а д елается в случ аях, когд а п ред став ление исп ользуется как замена самой исход ной таблицы, нап ример, если п ред став ление п росто исп ользует **CHECK** **OPTION** как заменитель огранич ения **CHECK** в базов ой таблице. Можно также созд ав ать собств енные синонимы п ользов ателя, имена которых буд ут такими же, как и п ерв онач альные имена таблиц. Н ап ример, п ользов атель PETROV может оп ред елить имя STUDENT как св ой синоним д ля таблицы IVANOV.STUDENT с п омощ ью зап роса:

**CREATE** **SYNONYM** STUDENT **FOR** IVANOV.STUDENT;

С точ ки зрения SQL, теп ерь имеются д в а разных имени од ной таблицы: IVANOV.STUDENT и PETROV.STUDENT. О д нако кажд ый из этих п ольз ов ателей может обращ аться к д анной таблице, исп ользуя имя STUDENT. SQL, как гов орилось в ыше, сам д обав ит к этому имени нед остающ ие имена п ользов ателей в кач еств е п рефиксов .

**6.11. С иноним ы общ его пол ь зования**

**(PUBLIC)**

Если п ланируется исп ользов ать таблицу STUDENT большим ч ислом п ольз ов ателей, уд обнее, ч тобы в се п ользов атели ссылались к ней с п омощ ью од ного и того же имени. Это д аст в озможность, нап ример, исп ользов ать указанное имя без огранич ений в п риклад ных п рограммах. Чтобы созд ать ед иное имя д ля в сех п ользов ателей, созд ается общ ий синоним.

Н ап ример, если в се п ользов атели буд ут в ыз ыв ать таблицу STUDENT с д анными о студ ентах, можно п рисв оить ей *о бщи й* синоним STUDENT след ующ им образом:

**CREATE** **PUBLIC** **SYNONYM** STUDENT **FOR** STUDENT;

О бщ ие синонимы в основ ном соз д аются в лад ельцами объектов или п ольз ов ателями с п рив илегиями ад министратора баз ы д анных (п ользов атель **DBA**). Другим п ольз ов ателям п ри этом д олжны быть п ред остав лены соотв етств ующ ие п рив илегии в таблице STUDENT, ч тобы она была им д оступ на. Даже если имя яв ляется общ им, сама таблица общ ей не яв ляется.

**6.12. Удал ение синонимов**

О бщ ие и д ругие синонимы могут уд аляться команд ой **DROP** **SYNONYM**. Синонимы могут уд аляться только их в лад ельцами, кроме общ их синонимов , которые могут уд аляться соотв етств ующ ими п рив илегиров анными п ольз ов ателями (обыч но это п ользов атели **DBA**). Чтобы уд алить, нап ример, синоним CLIENTS, когд а в место него уже п ояв ился общ ий синоним

STUDENT, п ользов атель может в в ести команд у

**DROP SYNONYM** CLIENTS;

7. Упр а влен и е тр а н за кци ями

В п роцессе в ып олнения п ослед ов ательности команд SQL таблицы базы д анных не в сегд а могут наход иться в согласов анном состоянии. В случ ае в озникнов ения каких-либо сбоев, когд а логич ески св яз анная п ослед ов ательность зап росов не д ов ед ена д о конца, в озможно нарушение целостности д анных в базе. Д ля обесп еч ения целостности д анных логич ески св яз анные п ослед ов ательности зап росов , нед елимые с точ ки з рения в озд ейств ия на баз у д анных, объед иняют в так назыв аемые *т р а нза кци и* . Зап росы, состав ляющ ие транзакцию, д олжны или в ып олняться в се п олностью – с п ерв ого д о п ослед него, и тогд а транз акция з ав ершается команд ой **COMMIT**, или, если в силу каких-либо в нешних п р ич ин это оказыв ается нев оз можным, в несенные зап росами транзакции изменения в базе д анных д олжны аннулиров аться команд ой **ROLLBACK**. Во в тором случ ае база д анных в оз в ращ ается в целостное состояние на момент, п ред шеств ующ ий нач алу транзакции. Это назыв а ют *о т ка т о м* транзакции.

Н ов ая транзакция нач инается п осле кажд ой команд ы **COMMIT** или **ROLLBACK**.

В большинств е реализа ций можно установ ить п араметр, назыв аемый **AUTOCOMMIT**. О н буд ет ав томатич ески зап оминать в се в ып олняемые д ейств ия над д анными. Действ ия, которые п рив ед ут к ошибке п ри незав ершенной транзакции, в сегд а буд ут ав томатич ески “откатаны” обратно.

Имеется в оз можность установ ки режима **AUTOCOMMIT** ав томатич ески п ри регистрации. Если сеанс п ользов ателя зав ершается ав арийно, нап ример, п роизошел сбой системы или в ып олнена п ерезагрузка п ользов ателя, то текущ ая транзакция в ып олнит ав томатич еский откат изменений. Это – од на из в озможностей уп рав ления в ып олнением д иалогов ой обработки зап росов п утем разд еления команд на большое колич еств о различ ных транзакций. О д иноч ная транзакция не д олжна сод ержать слишком много несв язанных команд , на п рактике она ч асто состоит из ед инств енной команд ы. Х орошее п рав ило, которому можно след ов ать – это созд ав ать транзакции из од ной команд ы или нескольких близко св язанных команд .

Н ап ример, требуется уд алить св ед ения о студ енте п о фамилии ‘Ива н о в’ из базы д анных. Прежд е, ч ем св ед ения из таблицы STUDENT буд ут уд алены, требуется осущ еств ить оп ред еленные д ейств ия с д анными об этом студ енте в д ругих таблицах, в ч астности с д анными о его оценках. Н еобход имо установ ить соотв етств ующ ее этому студ енту п оле STUDENT\_ID в таблице EXAM\_MARKS в **NULL**. После этого можно уд алить зап ись об этом студ енте из таблицы STUDENT. Эти д ейств ия в ып олняются с п омощ ью д в ух зап росов

**UPDATE** EXAM\_MARKS

**SET** STUDENT\_ID = **NULL**  **WHERE** STUDENT\_ID = 1004;

**DELETE** **FROM** STUDENT

**WHERE** STUDENT\_ID = 1004;

Если в озникает п роблема с уд алением з ап иси о студ енте с фамилией ‘Ива н о в’ (в озможно, имеется д ругой в нешний ключ , ссылающ ийся на него, о котором не было из в естно, и, соотв етств енно, не уч тено п ри уд алении), можно было бы отменить в се сд еланные изменения, п о крайней мере, д о тех п ор, п ока п роблема не буд ет решена. Для этого п рив ед енную груп п у команд след ует обрабатыв ать как од иноч ную транзакцию, п ред усматрив ая ее зав ершение с п омощ ью команд ы **COMMIT** или **ROLLBACK** – в зав исимости от результата.

**УП РАЖ НЕ НИ Я**

1. Вы п еред али п рав о **SELECT** в таблице EXAM\_MARKS п ользов ателю IVANOV. Вв ед ите команд у так, ч тобы в ы могли ссылаться к этой таблице, как к EXAM\_MARKS, не исп ользуя имя IVANOV в кач еств е п рефикса.
2. Если п роиз ойд ет сбой п итания, ч то случ ится со в семи изменениями, сд еланными в о в ремя текущ ей транзакции?

Пр едметн ы й ука за тель

DBA, 126

DDL, 4, 77 DML, 5 escape-симв ол, 20, 21

SQL

в строенный, 4 интерактив ный, 4

ад министратор баз ы д анных, 126 баз а д анных уч ебная, 10 таблица EXAM\_MARKS, 12 таблица LECTURER, 10 таблица STUDENT, 10 таблица SUBJ\_LECT, 12 таблица SUBJECT, 11 таблица UNIVERSITY, 11

в став ка

столбца, 79

строк, 68, 72

д екартов о п роизв ед ение, 58 изменение таблицы, 79, 83 инд ексация, 78

создание индекса, 79 удаление индекса, 79

исп ольз ов ание симв ола \*, 14 ключ

в нешний (FOREIGN KEY), 3, 60, 84, 91, 92,

93, 94, 95, 98, 100, 101 в озможный, 84

п ерв ич ный (PRIMERY KEY), 2, 83, 85, 86,

90, 96, 97 родительский, 91, 92, 93, 94, 97, 98 уникальный, 84

ключ ев ые слов а, 9 команд ы, 9

ALTER TABLE, 79, 80, 83, 93, 94, 95

ADD, 79

MODIFY, 80 добав ление столбца, 79 из менение описания столбцов , 80 синтаксис, 79, 85, 95

COMMIT, 131

CREATE INDEX, 79

CREATE TABLE, 68, 77, 82, 88, 93 синтаксис, 85

CREATE USER, 118, 127

CREATE VIEW, 102

DELETE, 68, 69, 73, 75, 98, 105, 123 огранич ение удаления род ительского

ключ а

CASCADE, 99

NO ACTION, 99

RESTRICT, 99

SET DEFAULT, 99

SET NULL, 99

DROP INDEX, 79 DROP TABLE, 80

DROP USER, 118

DROP VIEW, 112

GRANT, 118, 120, 122, 126

INSERT, 68, 69, 72, 77, 83, 98, 106, 123 VALUES, 68, 72 в став ить NULL-з нач ение, 68

REVOKE, 118, 122

ROLLBACK, 131

SELECT, 13, 78, 112

аргументы

ALL, 34

DISTINCT, 3, 15, 34 исп ользов ание симв ола \*, 14 оператор

JOIN, 59, 60 оператор объединения таблиц UNION, 13 в нешнее объединение, 57

п редложения

FROM, 13

FROM:, 13

GROUP BY, 34

GROUP BY, 13

HAVING, 13, 35, 42, 44

ORDER BY, 13, 39, 40, 56

ASC, 39

DESC, 39

WHERE, 13, 16, 19, 70, 72 синтаксис, 13

UPDATE, 68, 70, 75, 98, 104 огранич ение модификации

родительского ключ а

CASCADE, 99

NO ACTION, 99

RESTRICT, 99

SET NULL, 99

п редложение

SET, 70, 71, 116 синтаксис, 70

манип улиров ание данными, 68 оператор соединения таблиц JOIN, 58, 61, 62, 63

CROSS, 58

FULL OUTER JOIN, 63

INNER, 59, 61

LEFT OUTER JOIN, 62

RIGHT OUTER JOIN, 62

UNION JOIN, 63 логика трехзнач ная, 8, 37 манип улиров ание д анными, 68 мод ель д анных, 1 обнов ление, 70 обоз нач ения п ри оп исании синтаксиса

команд , 9

огранич ения, 81, 85

ALTER TABLE, 93

CHECK, 87, 125, 129

CONSTRAINT, 81, 85

CREATE TABLE, 93

DEFAULT, 83, 88, 89

DELETE, 98

FOREIGN KEY, 93, 94, 95, 98

INSERT, 98

NOT NULL, 81, 82

PRIMARY KEY, 86, 90, 96, 97

# UNIQUE, 83, 84, 85, 90

UPDATE, 98

WITH CHECK OPTION, 116, 124, 125, 129 альтернатив ы для NULL, 89 в командах

ALTER TABLE, 83, 85

CREATE TABLE, 77, 82, 85

INSERT, 83 в в од знач ений поля, 93

знач ения п о умолч анию, 88

ключ

в нешний (FOREIGN KEY), 93, 94, 95, 98

первич ный (PRIMERY KEY), 81, 83, 85,

86, 90, 96, 97 родительский, 91, 92, 93, 94, 97, 98, 99

модификация, 98

составной, 86, 92

модификация знач ений п оля, 93 п рисвоение имен, 85 п ров ерка знач ений полей, 87 ссылоч ная целостность, 91, 94, 97, 98 столбца, 81, 95

таблицы, 81, 82, 84, 86, 94, 95

удаление знач ений п оля, 93

удаления и модификации родительского

ключ а ON DELETE и ON UPDATE

CASCADE, 99

NO ACTION:, 99

RESTRICT, 99

SET DEFAULT, 99

SET NULL, 99

уникальность, 83, 84, 85, 90

оп ераторы – (в ыч итание), 23

\* (умножение), 23

/ (д еление), 23 || (конкатенация строк), 23

+ (сложение), 23

ALL, 50

ANY, 50

BETWEEN, 19, 20, 21, 117

COUNT, 52

EXISTS, 46, 50, 52

IN, 19, 42, 43, 48, 49, 73, 74, 75, 76, 115, 125

IS NOT NULL, 8

IS NULL, 8 LIKE, 19, 20, 21

NOT IN, 19, 48 UNION, 54, 56 конкатенация строк, 8 сравнение, 2, 16, 89

отмена п рив илегий, 122 отношение, 1 атрибут, 1 домен, 2 заголовок, 1 кардинальное ч исло, 2 ключ

в нешний (FOREIGN KEY), 3 первич ный (PRIMERY KEY), 2

кортеж, 1 св ойства, 2 степень, 2

п ароль, 118, 127

IDENTIFIED BY, 118, 127 п од зап росы, 73 в командах

DELETE, 73

UPDATE, 75 в предложениях

FROM, 74

HAVING, 42 в представ лениях, 111 в ложенные, 41 св язанные, 43, 46 в предложении HAVING, 44

п ольз ов атели, 118

создание, 118, 127 удаление, 118

п рав а д оступ а, см. п рив илегии, 118 п ред став ление (VIEW), 102, 104

агрегированное, 109 в став ка строки, 108 других п редстав лений, 116 защ ита данных, 114 использов ание

UNION и UNION ALL, 112 использов ание команд ы

DELETE, 105

GROUP BY, 109

INSERT, 68, 108, 123 маскирующ ее, 105

столбцы, 105, 108

модифицирование, 105 строки, 106 модифициров ание, 106, 108

многих таблиц, 110 модифицирование, 70, 75, 104

использов ание DISTINCT, 113

GROUP BY, 113

HAVING, 113 п одзапросы, 113

модифицирование знач ений, 113 не обновляемое, 113, 114 обновляемое, 113, 122 огранич ение исп ользов ания SELECT, 112 п одз апросы, 111 создание, 102 столбцов , 104 таблиц, 103 удаление, 112

п рефикс, 129 п рив илегии, 118, 120, 125

аргументы

ALL, 121

ALL PRIVILEGES, 121 PUBLIC, 121 базы данных, 126 в баз овых таблицах, 121 в п редстав лениях, 121 в иды, 119

ALTER, 119

DELETE, 119, 122

EXECUTE, 119

INDEX, 119 INSERT, 119, 122

REFERENCES, 119, 121, 123

SELECT, 119, 120, 123, 124

SYNONYM, 119

UPDATE, 119, 122, 123 в иды п ривилегий, 118 использ ов ание п редставлений, 122 огранич ение для строк, 123 отмена, 118 регистрации, 126 системы, 126

CONNECT, 126, 127

DBA, 126, 127

RESOURCE, 126, 127

А дминистратор Б аз ы Данных, 126

ПО Д К ЛЮ ЧИТЬ, 126

РЕСУ РС, 126 создав ать

базов ые таблицы, 126 п редставления, 126 синонимы, 126

установ ка, 118, 120, 126 фильтрация, 122 п сев д онимы, 64

реляционная мод ель д анных, 1 сбои, 130 симв ольные константы, 22 синонимы, 128 CREATE SYNONYM, 128

DROP SYNONYM, 130

общ его п ользов ания (PUBLIC), 130 создание, 128 удаление, 130

соед инение, 60 в нешнее, 61, 62 лев ое, 62 п олное, 63 п равое, 62 синтаксис ORACLE, 62

в нутреннее (INNER), 59, 61 использов ание псев д онимов , 64 п олное (CROSS), 58 эквисоединение, 59

созд ание

индексов , 79 объектов базы данных, 77 п ользователей, 118 п редстав лений, 102 синонимов , 128, 129 таблиц баз ы данных, 77

срав нение, 2, 16, 89 ссылоч ная целостность, 3, 60, 91, 92, 93,

94, 95, 97, 98, 99, 100, 101 станд арты ANSI, 77 столбец

добав ление, 79 из менение описания, 80

строка

в став ка, 68 идентификаторы строк ROWID, 78 удаление, 69, 75, 123

суп ерп ользов атель, 126 таблица, 1

базов ая, 102 в иртуальная, 104, 110 из менение, 79, 83 именов анная, 102 родительская, 94 удаление, 80

тип ы д анных, 5, 20, 21, 77

дата и время, 7

п роп ущ енные данные (NULL), 8, 36, 50, 68,

71, 81, 82, 85, 89 строка симв олов

CHAR, 5, 77

CHARVARYING, 6

CHARACTER, 5, 77

CHARACTER VARYING, 6

VARCHAR, 6

ч исловые типы, 6

DECIMAL, 6, 77

DOUBLE PRECISION, 7

FLOAT, 7, 77

INTEGER, 6, 77

NUMBER, 7

NUMERIC, 7, 77

REAL, 7

SMALLINT, 77

SMOLLINT, 6 транзакция, 130 AUTOCOMMIT, 131 зав ершение, 131 COMMIT, 131 ROLLBACK, 131 нормальное, 131 откат, 131

уд аление

индексов , 79 п ользов ателей, 118 п редставлений, 112 синонимов , 130 строк, 69, 75, 123 таблиц базы данных, 69, 80

функции агрегирующ ие, 33

AVG, 33, 37 COUNT, 33, 36

COUNT(\*), 34

MAX, 33

MIN, 33

SUM, 33

в строенные, 22

ABS, 27

CAST, 31

CEIL, 27

COS, 28

COSH, 28

EXP, 28

FLOOR, 27

INITCAP, 24 INSTR, 26

LENGTH, 26

LOWER, 24

LPAD, 24

LTRIM, 25

POWER, 28

ROUND, 27

RPAD, 25

RTRIM, 25

SIGN, 28

SIN, 28

SINH, 28

SQRT, 28

SUBSTR, 25

TAN, 28

TANH, 28

TO\_CHAR, 29

TO\_DATE, 30

TO\_NUMBER, 30

TRUNC, 27 UPPER, 24 п реобразование букв , 24 работы с ч ислами, 27 симв ольные строков ые, 24

целостность д анных, 131 экв исоед инение, 59 яз ык

обработки данных (DML), 5 определения данных (DDL), 4, 77

**П рил ож ение 1. З адачи по проектированию БД**

В п риложении 2 п р ив од ятся тексты з ад ач п о п роектиров анию баз д анных, относящ ихся к различ ным п ред метным областям. Требуется в соотв етств ии с услов иями зад ач :

* сформиров ать структуру таблиц баз д анных,
* п од обрать п од ход ящ ие имена таблицам и их п олям,
* обесп еч ить требов ания нормализации таблиц баз д анных (то есть п рив ед ение к п ятой нормальной форме),
* сформиров ать SQL зап росы д ля созд ания таблиц баз д анных с указ анием п ерв ич ных и в нешних ключ ей и необход имых огранич ений, SQL зап росы д ля д обав ления, из менения и в ыборки необход имых д анных.

При решении зад ач п ред п олагается исп ользов ание сред ств , п озв оляющ их разрабатыв ать схемы баз д анных, и п риложений, работающ их с баз ами д анных (Power Designer, Oracle Developer, ERWin, Power Builder, Borland Delphi, C++ Builder, и д р.)

**За да ча 1.** *Л ет о пи сь о ст р о ва Са нт а Бели нд а*

Г д е-то в в еликом океане лежит в оображаемый остров Санта Б елинд а . Вот уже триста лет в ед ется п од робная летоп ись остров а. В летоп ись заносятся и д анные обо в сех люд ях, хоть какое-то в ремя п рожив ав ших на остров е. Зап исыв аются их имена, п ол, д аты рожд ения и смерти. Х ранятся там и имена их род ителей, если изв естно, кто они. У некоторых отсутств уют св ед ения об отце, у некоторых – о матери, а ч асть люд ей, суд я п о зап исям, – круглые сироты. Из летоп иси можно уз нать, когд а был п остроен кажд ый д ом, стоящ ий на остров е (а если сейч ас его уже нет, то когд а он был снесен), точ ный ад рес и п од робный п лан этого д ома, кто и когд а в нем жил.

Точ но так же, как и столетия назад , на остров е д ейств уют п ред п риниматели, занима ющ иеся, в ч астности, лов лей рыбы, заготов кой сахарного тростника и табака. Б ольшинств о из них в се д елают сами, а некоторые нанимают работников , заключ ая с ними контракты раз ной п род олжительности. Имеются зап иси и о том, кто кого нанимал, на какую работу, когд а нач ался и законч ился контракт. Собств енно, круг занятий жителей остров а крайне нев елик и не меняется в еками. Н еуд ив ительно п оэтому, ч то в летоп иси п од робно оп исыв ается кажд ое д ело, буд ь то рыбная лов ля или в ып еч ка хлеба. Все п ред п риниматели – уроженцы остров а. Н екоторые объед иняются в кооп ератив ы, и п о з ап исям можно установ ить, кто уч аств ов ал в д еле, когд а в ступ ил и когд а в ышел из него, каким п аем в лад ел. Имеются краткие оп исания д еятельности кажд ого п ред п ринимателя или кооп ератив а, сообщ ающ ие, в том ч исле, когд а было нач ато д ело, когд а и п оч ему п рекращ ено.

Пред лагается сформиров ать систему нормализ ов анных таблиц, в которых можно было бы хранить в сю эту многообразную информацию. Под ыщ ите в ыраз ительные имена д ля таблиц и п олей, снабд ив их п ри необход имости соотв етств ующ ими п ояснениями.

**За да ча 2.** *Ба за д а нны х “Ска чки ”.*

В информационной системе клуба любителей скач ек д олжна быть п ред став лена информация об уч аств ующ их в скач ках лошад ях (клич ка, п ол, в озраст), их в лад ельцах (имя, ад рес, телефон) и жокеях (имя, ад рес, в оз раст, рейтинг). Н еобход имо сформиров ать таблицы д ля хранения информации п о кажд ому состязанию: д ата, в ремя и место п ров ед ения скач ек (ип п од ром), назв ание состязаний (если таков ое имеется), номера заезд ов , клич ки уч аств ующ их в заез д ах лошад ей и имена жокеев, занятые ими места и п оказ анное в заез д е в ремя.

**За да ча 3.** *Ба за д а нны х «Х р о ни ки во схо ж д ени й» в а льпи ни ст ско м клубе.*

В базе д анных д олжны зап исыв аться д аты нач ала и зав ершения кажд ого в осхожд ения, имена и ад реса уч аств ов ав ших в нем альп инистов , назв ание и в ысота горы, страна и район, гд е эта гора расп оложена. Присв ойте в ыразительные имена таблицам и п олям д ля хра нения указанной информации. Н ап исать зап росы, осущ еств ляющ ие след ующ ие оп ерации:

1. Для в в ед енного п ользов ателем интерв ала д ат п оказать сп исок гор с указ анием д аты п ослед него в осхожд ения. Д ля кажд ой горы сформиров ать в хронологич еском п оряд ке сп исок груп п , осущ еств ляв ших в осхожд ение.
2. Пред остав ить в озможность д обав ления нов ой в ершины с указанием ее назв а ния, в ысоты и страны местоп оложения.
3. Пред остав ить в озможность изменения д анных о в ершине, если на нее не было в осхожд ения.
4. Показать сп исок альп инистов , осущ еств ляв ших в осхожд ение в указанный интерв ал д ат. Для кажд ого альп иниста в ыв ести сп исок гор, на которые он осущ еств лял в осхожд ения в этот п ериод , с указанием назв ания груп п ы и д аты в осхожд ения.
5. Пред остав ить в озможность д обав ления нов ого альп иниста в состав указанной груп п ы.
6. Показать информацию о колич еств е в осхожд ений кажд ого альп иниста на кажд ую гору. При в ыв од е сп исок отсортиров ать п о колич еств у в осхожд ений.
7. Показать сп исок в осхожд ений (груп п ), которые осущ еств лялись в указанный п ольз ов ателем п ериод в ремени. Д ля кажд ой груп п ы п оказ ать ее состав .
8. Пред остав ить в озможность д обав ления нов ой груп п ы, указав ее назв ание, в ершину, в ремя нач ала в осхожд ения.
9. Пред остав ить информацию о том, сколько альп инистов п обыв али на кажд ой горе. Сп исок отсортиров ать в а лфав итном п оряд ке п о назв анию в ершин.

**За да ча 4.** *Ба за д а нны х м ед и ци нского коо пер а т и ва .*

Б азу д анных исп ользует д ля работы коллектив в рач ей. В таблицы д олжны быть занесены имя, п ол, д ата рожд ения и д омашний ад рес кажд ого их п ациента. Всякий раз , когд а в рач осматрив ает больного (п ришед шего на п рием или на д ому), фиксируется д ата и место п ров ед ения осмотра, симп томы, д иагноз и п ред п исания больному, п ростав ляется имя п ациента и имя в рач а. Если в рач п роп исыв ает больному какое-либо лекарств о, в таблицу заносится назв ание лекарств а, сп особ его п риема, слов есное оп исание п ред п олагаемого д ейств ия и в оз можных п обоч ных эффектов .

**За да ча 5.** *Ба за д а нны х «Го р о д ска я Д ум а ».*

В баз е хранятся имена, ад реса, д омашние и служебные телефоны в сех ч ленов Д умы. В Думе работает п оряд ка сорока комиссий, в се уч астники которых яв ляются ч ленами Думы. К ажд ая комиссия имеет св ой п рофиль, нап ример, в оп росы образов ания, п роблемы, св яз анные с жильем, и так д алее. Да нные п о кажд ой из комиссий в ключ ают: п ред сед атель и состав , п режние (з а 10 п ред ыд ущ их лет) п ред сед атели и ч лены этой комиссии, д аты в ключ ения и в ыход а из состав а комиссии, из брания ее п ред сед ателей. Члены Д умы могут засед ать в нескольких комиссиях. В базу заносятся в ремя и место п ров ед ения кажд ого засед ания комиссии с указанием д еп утатов и служащ их Думы, которые уч аств уют в его организ ации.

1) Показать сп исок комиссий, д ля кажд ой – ее состав и п ред сед ателя. 2) Пред остав ить в оз можность д обав ления нов ого ч лена комиссии.

1. Для в в ед енного п ользов ателем интерв ала д ат и назв а ния комиссии п оказ ать в хронологич еском п оряд ке в сех ее п ред сед ателей.
2. Показать сп исок ч ленов Д умы, д ля кажд ого из них – сп исок комиссий, в которых он уч аств ов ал и/или был п ред сед ателем.
3. Пред остав ить в озможность д обав ления нов ой комиссии, с указанием п ред сед ателя.
4. Для указанного интерв ала д ат и комиссии в ыд ать сп исок ч ленов с указанием колич еств а п роп ущ енных засед аний.
5. Выв ести сп исок засед аний в указанный интерв ал д ат в хронологич еском п оряд ке, д ля кажд ого засед ания – сп исок п рисутств ующ их.
6. Пред остав ить в озможность д обав ления нов ого з асед ания, с указанием п рисутств ующ их.
7. По кажд ой комиссии п оказать колич еств о п ров ед енных засед аний в указанный п ериод в ремени.

**За да ча 6.** *Ба за д а нны х р ы бо ло вно й ф и р м ы .*

Фирме п ринад лежит небольша я флотилия рыболов ных катеров . К ажд ый катер имеет “п асп орт”, куд а занесены его наз в ание, тип , в од оиз мещ ение и д ата п остройки. Фирма регистрирует кажд ый в ыход на лов , з ап исыв ая наз в ание катера, имена и ад реса ч ленов команд ы с указанием их д олжностей (кап итан, боцман и т.д .), д аты в ыход а и в оз в ращ ения, а также в ес п ойманной рыбы отд ельно п о сортам (нап ример, трески). За в ремя од ного рейса катер может п осетить несколько рыболов ных мест (банок). Фиксируется д ата п риход а на кажд ую банку и д ата отп лытия, кач еств о в ылов ленной рыбы (отлич ное, хорошее, п лохое). Н а борту улов не в зв ешив ается. Н ап исать зап росы, осущ еств ляющ ие след ующ ие оп ерации:

* 1. По указанному тип у и интерв алу д ат в ыв ести в се катера, осущ еств ляв шие в ыход в море, указав д ля кажд ого в хронологич еском п оряд ке з ап иси о в ыход е в море и знач ением улов а.
  2. Пред остав ить в озможность д обав ления в ыход а катера в море с указанием кома нд ы.
  3. Для указанного интерв ала д ат в ыв ести д ля кажд ого сорта рыбы сп исок катеров с на ибольшим улов ом.
  4. Для указанного интерв ала д ат в ыв ести сп исок ба нок, с указанием сред него улов а з а этот п ериод . Для кажд ой банки в ыв ести сп исок катеров , осущ еств ляв ших лов .
  5. Пред остав ить в оз можность д обав ления нов ой банки с указанием д а нных о ней.
  6. Для зад анной банки в ыв ести сп исок катеров , которые п олуч или улов в ыше сред него.
  7. Выв ести сп исок сортов рыбы и д ля кажд ого сорта – сп исок рейсов с указанием д аты в ыход а и в озв ращ ения, в елич ины улов а. При этом сп исок п оказанных рейсов д олжен быть огранич ен интерв алом д ат.
  8. Для в ыбранного п ользов ателем рейса и банки д обав ить д анные о сорте и колич еств е п ойманной рыбы.
  9. Пред остав ить в озможность п ользов ателю изменять характеристики в ыбранного катера.
  10. Для указанного интерв ала д ат в ыв ести в хронологич еском п оряд ке сп исок рейсов за этот п ериод в ремени, с указанием д ля кажд ого рейса п ойма нного колич еств а кажд ого сорта рыбы.
  11. Пред остав ить в озможность д обав ления нов ого катера.
  12. Для указ анного сорта рыбы и банки в ыв ести сп исок рейсов с указанием колич еств а п ойманной рыбы. Сп исок д олжен быть отсортиров ан в п оряд ке уменьшения колич еств а п ойманной рыбы.

**За да ча 7.** *Ба за д а нны х ф и р м ы , пр о во д ящей а укци о ны .*

Фирма занимается п род ажей с аукциона антикв арных изд елий и п роизв ед ений искусств а. Влад ельцы в ещ ей, в ыстав ляемых на п ров од имых фирмой аукционах, юрид ич ески яв ляются п род ав цами. Л ица, п риобретающ ие эти в ещ и, именуются п окуп ателями. Получ ив от п род ав цов п артию п ред метов , фирма решает, на котором из аукционов в ыгод нее п ред став ить конкретный п ред мет. Перед п ров ед ением оч еред ного аукциона кажд ой из в ыстав ляемых на нем в ещ ей п рисв аив ается отд ельный номер лота. Д в е в ещ и, п род ав аемые на различ ных аукционах, могут иметь од инаков ые номера лотов .

В книгах фирмы д елается з ап ись о кажд ом аукционе. Там отмеч аются д ата, место и в ремя его п ров ед ения, а также сп ецифика (нап ример, в ыстав ляются картины, нап исанные маслом и не ранее 1900 г.). Заносятся также св ед ения о кажд ом п род ав аемом п ред мете: аукцион, на который он заяв лен, номер лота, п род ав ец, отп рав ная цена и краткое слов есное оп исание. Прод ав цу разрешается в ыстав лять любое колич еств о в ещ ей, а п окуп атель имеет п рав о п риобретать любое колич еств о в ещ ей. О д но и то же лицо или фирма может в ыступ ать и как п род ав ец, и как п окуп атель. После аукциона служащ ие фирмы, п ров од ящ ей аукционы, зап исыв ают фактич ескую цену, уп лач енную за п род анный п ред мет, и фиксируют д анные п окуп ателя.

Н ап исать з ап росы, осущ еств ляющ ие след ующ ие оп ерации:

1. Для указанного интерв ала д ат в ыв ести сп исок аукционов в хронологич еском п оряд ке с указанием на именов ания, д аты и места п ров ед ения. Для кажд ого из них п оказать сп исок в ыстав ленных в ещ ей.
2. Добав ить д ля п род ажи на указ анный п ользов ателем аукцион п ред мет искусств а с указанием нач а льной цены.
3. Выв ести сп исок аукционов с указанием отсортиров анных п о в елич ине суммарных д оход ов от п род ажи.
4. Для указанного интерв ала д ат в ыв ести сп исок п род анных на аукционах п ред метов . Для кажд ого из п ред метов д ать сп исок аукционов , гд е в ыстав лялся этот же п ред мет.
5. Пред остав ить в озможность д обав ления факта п род ажи на указанном аукционе зад анного п ред мета.
6. Для указанного интерв ала д ат в ыв ести сп исок п род ав цов в п оряд ке убыв ания общ ей суммы, п олуч енной ими от п род ажи п ред метов в этот п ромежуток в ремени.
7. Выв ести сп исок п окуп ателей и д ля кажд ого из них – сп исок аукционов , гд е были сд еланы п риобретения в указанный интерв ал д ат.
8. Пред остав ить в оз можность д обав ления зап иси о п ров од имом аукционе (место, в ремя).
9. Для указ анного места в ыв ести сп исок аукционов , отсортиров анных п о колич еств у в ыста в ленных в ещ ей.
10. Для указанного интерв ала д ат в ыв ести сп исок п род ав цов , которые п ринимали уч астие в аукционах, с указанием д ля кажд ого из них сп иска в ыста в ленных п ред метов .
11. Пред остав ить в оз можность д обав ления и изменения информации о п род ав цах и п окуп ателях.
12. Выв ести сп исок п окуп ателей с указанием колич еств а п риобретенных п ред метов в указанный п ериод в ремени.

**За да ча 8.** *Ба за д а нны х м узы ка льно го м а га зи на .*

Таблицы базы д анных сод ержат информацию о муз ыканта х, музыкальных п роизв ед ениях и обстоятельств ах их исп олнения. Н ескольких музыкантов , образующ их ед иный коллектив , назыв а ются ансамблем. Это может быть классич еский оркестр, д жазов ая груп п а, кв артет, кв интет и т.д . К муз ыкантам п рич исляют исп олнителей (играющ их на од ном или нескольких инструментах), комп озиторов , д ирижеров и руков од ителей ансамблей.

К роме того, в баз е д анных хранится информация о комп актд исках, которыми торгует магазин. К ажд ый комп акт-д иск, а точ нее, его наклейка, ид ентифицируется отд ельным номером, так ч то в сем его коп иям, созд анным в разное в ремя, п рисв оены од инаков ые номера. Н а комп акт-д иске может быть зап исано несколько в ариантов исп олнения од ного и того же п роиз в ед ения – д ля кажд ого из них в базе зав ед ена отд ельная зап ись. К огд а в ыход ит нов ый комп акт-д иск, регистрируется наз в ание в ып устив шей его комп ании (нап ример, EMI), а также ад рес оп тов ой фир мы, у которой магазин может п риобрести этот комп акт-д иск. Н е исключ ено, ч то комп ания–п роизв од итель занимается и оп тов ой п род ажей комп акт-д исков . Магазин фиксирует текущ ие оп тов ые и рознич ные цены на кажд ый комп акт-д иск, д ату его в ып уска, колич еств о экз емп ляров , п род анных за п рошлый год и в нынешнем год у, а также ч исло ещ е не п род анных комп акт-д исков .

**За да ча 9.** *Ба за д а нны х кегельно й ли ги .*

Став ится з ад ач а сп роектиров ать баз у д анных д ля секретаря кегельной лиги небольшого город ка, расп оложенного на Сред нем Зап ад е СШ А . В ней секретарь буд ет хранить в сю информацию, относящ уюся к кегельной лиге, а сред ств ами СУ Б Д – формиров ать еженед ельные отч еты о состоянии лиги. Сп ециальный отч ет п ред п олагается формиров ать в конце сезона.

Секретарю п онад обятся фамилии и имена ч ленов лиги, их телефонные номера и ад реса. Так как в лигу могут в ход ить только жители город ка, нет необход имости хранения д ля кажд ого игрока назв ания город а и п оч тов ого инд екса. Интерес п ред став ляют ч исло оч ков , набранных кажд ым игроком в еженед ельной серии из трех в стреч , в которых он п ринял уч астие, и его текущ ая результатив ность (сред нее ч исло набираемых оч ков в од ной в стреч е). Секретарю необход имо з нать д ля кажд ого игрока назв ание команд ы, за которую он в ыступ ает, и фамилию (и имя) кап итана кажд ой команд ы. Помимо наз в ания, секретарь п ланирует назнач ить кажд ой команд е уникальный номер.

Исход ные знач ения результатив ности кажд ого игрока необход имы как п ри оп ред елении в конце сезона д остигшего наибольшего п рогресса в лиге игрока, так и п ри в ыч ислении ганд икап а д ля кажд ого игрока на п ерв ую нед елю нов ого сезона. Л уч шая игра кажд ого игрока и луч шие серии п отребуются п ри расп ред елении п ризов в конце сезона.

Секретарь п ланирует в ключ ать в еженед ельные отч еты информацию об общ ем ч исле набранных оч ков и общ ем ч исле п ров ед енных игр кажд ым игроком, эта информация исп ользуется п ри в ыч ислении их текущ ей результатив ности и текущ его ганд икап а. Исп ользуемый в лиге ганд икап состав ляет 75% от разности межд у 200 и результатив ностью игрока, п ри этом отрицательный ганд икап не д оп ускается. Если результатом в ыч исления га нд икап а яв ляется д робная в елич ина, то она усекается. Перерасч ет ганд икап а осущ еств ляется кажд ую нед елю.

Н а кажд ую нед елю кажд ой команд е требуется назнач ать п лощ ад ку, на которой она буд ет в ыступ ать. Эту информацию хранить в Б Д не нужно (соп ерники в ыступ ают на смежных п лощ ад ках).

Н аконец, в Б Д д олжна сод ержаться в ся информация, необход имая д ля расч ета п оложения команд . К ома нд е засч итыв ается од на п обед а за кажд ую игру, в которой ей уд алось набрать больше оч ков (в ыбить больше кеглей) (с уч етом га нд икап а), ч ем команд е соп ерников . Точ но также команд е засч итыв ается од но п оражение за кажд ую в стреч у, в которой эта команд а в ыбила меньшее колич еств о кеглей, ч ем команд а соп ерников . К оманд е также засч итыв ается од на п обед а (п оражение) в случ ае, если п о срав нению с команд ой соп ерников ею набрано больше (меньше) оч ков за три в стреч и, состояв шиеся на нед еле. Таким образом, на кажд ой нед еле разыгрыв ается 4 команд ных оч ка (п обед или п оражений). В случ ае нич ейного результата кажд ая команд а п олуч ает 1/2 п обед ы и 1/2 п оражения. В случ ае неяв ки более ч ем д в ух ч ленов команд ы, их команд е ав томатич ески з асч итыв ается 4 п оражения, а команд е соп ерников – 4 п обед ы. В общ ий результат команд е, которой засч итана неяв ка , оч ки не п рибав ляются, д аже если яв ив шиеся игроки в этой в стреч е в ыступ или, од нако, в инд ив ид уальные п оказатели – ч исло набранных оч ков и п ров ед енных в стреч – буд ут в несены соотв етств ующ ие изменения. Н ап исать з ап росы, осущ еств ляющ ие след ующ ие оп ерации:

1. Для указанного интерв ала д ат п оказ ать сп исок в ыступ ающ их команд . Д ля кажд ой из них в ыв ести состав и кап итана команд ы.
2. Пред остав ить в оз можность д обав ления нов ой команд ы.
3. Выв ести сп исок игров ых п лощ ад ок с указанием колич еств а п ров ед енных игр на кажд ой их них.
4. Для указанного интерв ала д ат в ыв ести сп исок игров ых п лощ ад ок, с указанием сп иска играв ших на них команд .
5. Пред остав ить в озможность з ап олнения результатов игры д в ух команд на указанной п лощ ад ке.
6. Выв ести сп исок п лощ ад ок с указанием суммарной результатив ности игроков на кажд ой из них.

**За да ча 10.** *Ба за д а нны х би бли о т еки .*

Разработать информационную систему обслужив а ния библиотеки, которая сод ержит след ующ ую информацию: назв а ние книги, ФИО ав торов , наименов ание из д ательств а, год из д ания, колич еств о страниц, колич еств о иллюстраций, стоимость, наз в ание филиала библиотеки или книгохранилищ а, в которых наход ится книга, колич еств о имеющ ихся в библиотеке экземп ляров конкретной книги, колич еств о студ ентов , которым в ыд а в алась конкретная книга , назв а ния факультетов , в уч ебном п роцессе которых исп ользуется указанная книга.

**За да ча 11.** *Ба за д а нны х по учет у успева ем о ст и ст уд ент о в.*

Б аза д анных д олжна сод ержать д анные:

о контингенте студ ентов – фамилия, имя, отч еств о, год п оступ ления, форма обуч ения (д нев ная/в еч ерняя/з аоч ная), номер или назв а ние груп п ы; об уч ебном п лане – назв а ние сп ециальности, д исцип лина, семестр, колич еств о отв од имых на д исцип лину ч асов , форма отч етности

(экз амен/зач ет); о журнале усп ев аемости студ ентов – год /семестр, студ ент, д исцип лина, оценка.

**За да ча 12.** *Ба за д а нны х д ля учет а а уд и т о р ного ф о нд а уни вер си т ет а .*

Б аза д анных д олжна сод ержать след ующ ую информацию об ауд иторном фонд е унив ерситета : – наименов ание корп уса, в котором расп оложено п омещ ение, номер комнаты, расп оложение комнаты в корп усе, ширина и д лина комнаты в метрах, назнач ение и в ид п омещ ения, п од разд еление унив ерситета, за которым закреп лено п омещ ение. В баз е д анных также д олжна быть информация о в ысоте п отолков в п омещ ениях (в зав исимости от места расп оложения п омещ ений в корп усе). След ует также уч итыв ать, ч то структура п одразд елений унив ерситета имеет иерархич еский в ид , когд а од ни п од раз д еления в ход ят в состав д ругих (факультет, кафед ра, лаборатория, … ).

Помимо SQL з ап росов д ля соз д ания таблиц базы д анных, состав ьте зап рос на созд ание п ред став ления (**VIEW**), в котором п омимо п рив ед енной в ыше информации п рисутств ов а ли бы д анные о п лощ ад ях и объемах кажд ого п омещ ения.

**За да ча 13.** *Ба за д а нны х р еги ст р а ци и пр ои сш ест ви й.*

Н еобход имо созд ать базу д анных регистрации п роисшеств ий. Б аза д олжна сод ержать:

* д анные д ля регистрации сообщ ений о п роисшеств иях (регистрационный номер сообщ ения, д ата регистрации, краткая фабула (тип п роисшеств ия));
* информацию о п ринятом п о п роисшеств ию решении (отказано

в в оз бужд ении д ел, уд ов летв орено ход атайств о о в оз бужд ении уголов ного д ела с указанием регистрационный номера з ав ед енного д ела, отп рав лено п о территориальному п риз наку);

* информацию о лицах, в инов ных или п од озрев аемых в сов ершении п роисшеств ия (регистрационный номер лица, фамилия, имя, отч еств о, ад рес, колич еств о суд имостей), отношение конкретных лиц к конкретным п роисшеств иям (в инов ник, п отерп ев ший, п од оз рев аемый, св ид етель, ...).

**За да ча 14.** *Ба за д а нны х д ля о бслуж и ва ни я р а бо т ы ко нф ер енци и .*

Б аза д анных д олжна сод ержать сп рав оч ник п ерсоналий уч астников конференции (фамилия, имя, отч еств о, уч еная степ ень, уч еное з в ание, науч ное на п рав ление, место работы, кафед ра (отд ел), д олжность, страна, город , п оч тов ый инд екс, ад рес, рабоч ий телефон, д омашний телефон, e-mail), и информацию, св яз анную с уч астием в конференции (д оклад ч ик или уч астник, д ата рассылки п ерв ого п риглашения, д ата п оступ ления заяв ки, тема д оклад а, отметка о п оступ лении тез исов , д ата рассылки в торого п риглашения, д ата п оступ ления оргв зноса, размер п оступ ив шего оргв з носа, д ата п риез д а, д ата отъезд а, п отребность в гостинице).

**За да ча 15.** *Ба за д а нны х д ля о бслуж и ва ни я скла д а .*

Б аза д анных д олжна обесп еч ить ав томатизацию склад ского уч ета. В ней д олжны сод ержаться след ующ ие д анные:

* информация о “ед иницах хранения” – номер орд ера, д ата, код п остав щ ика, балансный сч ет, код соп ров од ительного д окумента п о сп рав оч нику д окументов , номер соп ров од ительного д окумента, код материа ла п о сп рав оч нику материалов , сч ет материала, код ед иницы из мерения, колич еств о п ришед шего материала, цена ед иницы измерения;
* информация о хранящ ихся на склад е материалах – сп ра в оч ник материалов – код класса материала, код груп п ы материала, наименов ание материала;
* информация о ед иницах из мерения конкретных в ид ов материалов – код материала, ед иница из мерения (метры, килограммы, литры и т.д .).
* информация о п остав щ иках материа лов – код п остав щ ика, его наименов ание, ИН Н , юрид ич еский ад рес (инд екс, город , улица, д ом), ад рес банка (инд екс, город , улица , д ом), номер банков ского сч ета;

**За да ча 16.** *Ба за д а нны х ф и р м ы .*

Фирма отказ алась от п риобретения некоторых тов аров у св оих п остав щ иков , решив самостоятельно на лад ить их п роизв од ств о. С этой целью она организов ала сеть сп ециализиров анных цехов , кажд ый из которых п ринимает оп ред еленное уч астие в технологич еском п роцессе.

К ажд ому в ид у в ып ускаемой п род укции п рисв аив ается, как обыч но, св ой шифр тов ара, п од которым он знач ится в файле тов арных зап асов . Этот же номер служит и шифром п род укта. В зап иси с этим шифром указыв ается, когд а была из готов лена п ослед няя п артия этого п род укта, каков а ее стоимость, сколько оп ераций п отребов алось.

О п ерацией сч итается з аконч енная ч асть п роцесса п роизв од ств а, которая целиком в ып олняется силами од ного цеха в соотв етств ии с технич ескими требов аниями, п ереч исленными на отд ельном ч ертеже. Для кажд ого п род укта и д ля кажд ой оп ерации в базе д анных фирмы зав ед ена з ап ись, сод ержащ ая оп исание оп ерации, ее сред нюю п род олжительность и номер ч ертежа, п о которому можно отыскать требуемый ч ертеж. К роме того, указыв ается номер цеха, обыч но п роизв од ящ его д анную оп ерацию.

В зап ись, св язанную с конкретной оп ерацией, заносятся п отребные колич еств а расход ных материа лов , а также п рисв оенные им шифры тов ара. Расход ными наз ыв а ют такие материалы, ка к, нап ример, электрич еский кабель, который нельзя исп ользов ать п ов торно. При в ыд ач е расход ного материала со склад а в п роцессе п од готов ки к в ып олнению оп ерации, регистрируется фактич ески в ыд анное колич еств о, соотв етств ующ ий шифр тов ара, номер служащ его, отв етств енного за в ыд ач у, д ата и в ремя в ыд ач и, номер оп ерации и номер наряд а на п ров ед ение работ (о котором несколько ниже). Реально затрач енное колич еств о материала может не сов п ад ать с расч етным, (нап ример, из-з а брака).

К ажд ый из цехов расп олагает требуемым инструментарием и оборуд ов анием. При в ып олнении некоторых оп ераций их иногд а нед остаточ но, и цех в ынужд ен обращ аться в центральную инструментальную за нед остающ ими. К ажд ый тип инструмента снабжен отд ельным номером и на него зав ед ена зап ись со слов есным оп исанием. К роме того, отмеч ается, какое колич еств о инструментов этого тип а в ыд елено цехам и какое осталось в инструментальной. Э кз емп ляры инструмента конкретного тип а, нап ример, гаеч ные ключ и од ного размера, различ аются п о св оим инд ив ид уальным номерам. Н а фирме д ля кажд ого тип а инструмента имеется зап ись, сод ержащ ая п ереч ень в сех инд ив ид уальных номеров . К роме того, указаны д аты их п оступ ления на склад .

По кажд ой оп ерации в фирме отмеч ают тип ы и колич еств а инструментов этих тип ов , которые д олжны исп ользов аться п ри ее в ып олнении. К огд а инструменты д ейств ительно берутся со склад а, фиксируется инд ив ид уальный номер кажд ого экз емп ляра, указ ыв аются номер заказав шего их цеха и номер наряд а на п ров ед ение работ. И в этом случ ае затребов анное колич еств о не в сегд а сов п ад ает с з аказ анным.

Н аряд на п ров ед ение работ п о форме нап оминает заказ на п риобретение тов аров , но, в отлич ие от п ослед него, нап рав ляется не п остав щ ику, а в од ин из цехов . О формляется наряд п осле того, как руков од ств о фирмы соч тет необход имым в ып устить п артию некоторого п род укта. В наряд з аносятся шифр п род укта, д ата оформления наряд а, срок, к которому д олжен быть в ып олнен заказ, а также требуемое колич еств о п род укта.

Разработайте структуру таблиц баз ы д анных, п од берите имена таблиц и п олей, в которых могла бы разместиться в ся эта информация.

Н ап ишите SQL-зап росы, осущ еств ляющ ие след ующ ие оп ерации:

1. Для в ыбранного цеха в ыд ать сп исок в ып олняемых им оп ераций. Для кажд ой оп ерации п оказ ать сп исок расход ных материалов с указанием колич еств а.
2. Показать сп исок инструментов и п ред остав ить в оз можность д обав ления нов ого.
3. Выд ать сп исок исп ольз уемых инструментов , отсортиров анных п о колич еств у их исп ольз ов ания в различ ных наряд ах.
4. Для указанного интерв ала д ат в ыв ести сп исок наряд ов в хронологич еском п оряд ке, д ля кажд ого из которых п оказ ать сп исок исп ользуемых инструментов .
5. Показать сп исок оп ераций и п ред остав ить в оз можность д обав ления нов ой оп ерации.
6. Выд ать сп исок расход уемых материа лов , отсортиров анных п о колич еств у их исп ольз ов ания в различ ных наряд ах.
7. Выд ать сп исок тов аров , с указанием исп ользуемых п ри их изготов лении инструментов .
8. Показать сп исок наряд ов в хронологич еском п оряд ке и п ред остав ить в озможность д обав ления нов ого.
9. Выд ать отч ет о п роизв од ств е тов аров различ ными цехами, указав наименов ание цеха, наз в ание тов ара и его колич еств о.

Л и тер а тур а

1. Дейт К . Вв ед ение в системы баз д анных. 6-е изд ание: Пер. с англ. – К .; М.; СПб.: Изд ательский д ом “Вильямс”, 1999.- 848 с., ил.
2. М артин Г рабер. Вв ед ение в SQL. – М.: “Л О РИ”, 1996
3. SQL. Э нциклоп ед ия п ольз ов ателя: Пер. с англ./ Лад ани Х анс.- К иев : ДиаСофт, 1998.- 624 с.
4. Г .Джексон. Проектиров ание реляционных баз д анных д ля исп ольз ов ания с микро-ЭВМ.- М .: М ир, 1991
5. У льма н Дж. Б азы д анных на Паскале. - М .: М ашиностроение, 1990. – 462 с.
6. Пушников А .Ю . Вв ед ение в системы уп рав ления баз ами д анных. Часть 1. Реляционная мод ель д анных: У ч ебное п особие/ Изд -е Б ашкирского ун-та. – У фа. 1999. – 108 с.
7. Пушников А .Ю . Вв ед ение в системы уп рав ления баз ами д анных. Часть 2.

Н ормальные формы отношений и транзакции: У ч ебное п особие/ Из д -е Б ашкирского ун-та. – У фа. 1999. – 108 с.