МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Череповецкий государственный университет»

**Лабораторная работа № 2**

ПОДДЕРЖАНИЕ ЦЕЛОСТНОСТИ ДАННЫХ

**Выполнил:**

студент гр. 1ИВТпб-01-41оп

Климов А.Г.

**Проверил:** преподаватель

Селяничев О.Л.

Отметка о зачете:

Череповец

2019 год

Указания к выполнению

1. Ознакомиться с теорией вопроса.
2. Изучить примеры.
3. Предложить свои задания по данному вопросу на основе предметной области «Студенческая жизнь».
4. Решить задание.

В отчете отразить:

* тематику вопроса;
* придуманное задание по этой теме;
* решения (или оба).

1. Преимущества применения ограничения целостности

Указания к выполнению

Решить это задание, применяя

а) элементы ограничения целостности и

б) операторы SQL (предложения SQL) без привлечения элементов ограничения целостности;

сравнить, оценить достоинства того или другого решения.

**Ход работы:**

Определить ограничение целостности. **Ограничение целостности** – это правило, которое ограничивает допустимые значения одного или нескольких столбцов в таблице.

1.1. Ограничение PRIMARY KEY

**PRIMARY KEY** — первичный ключ, ограничение, позволяющее однозначно идентифицировать каждую запись в таблице **SQL**.

Первичный Ключ (**PRIMARY KEY**) может ограничивать таблицы или их столбцы. Это ограничение работает так же как и ограничение UNIQUE. Но следует учитывать различие между первичными ключами и уникальностью столбцов в способе их использования с внешними ключами. Первичные ключи не могут позволять значений NULL. Это означает что, подобно полям в ограничении UNIQUE, любое поле, используемое в ограничении **PRIMARY KEY**, должно уже быть обьявлено NOT NULL.

**Задание:**

Создать таблицу студентов с числовым идентификатором, фамилией и адресом.

**Решение 1:**

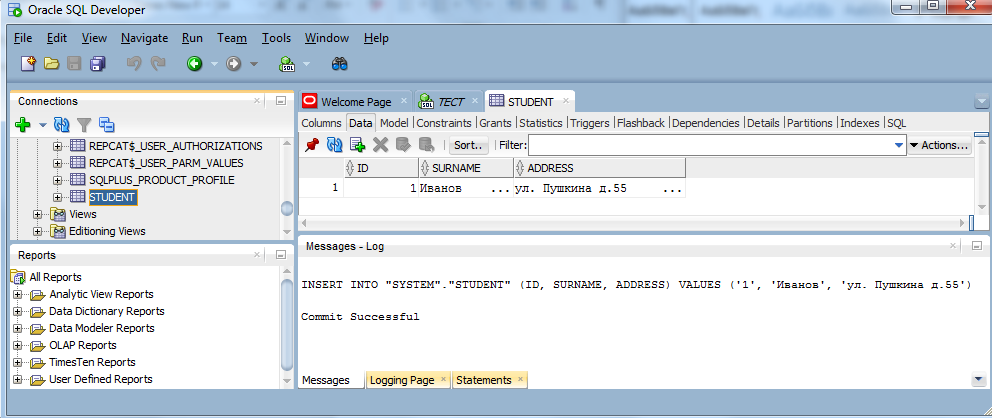
CREATE TABLE Student

(Id integer NOT NULL PRIMARY KEY,

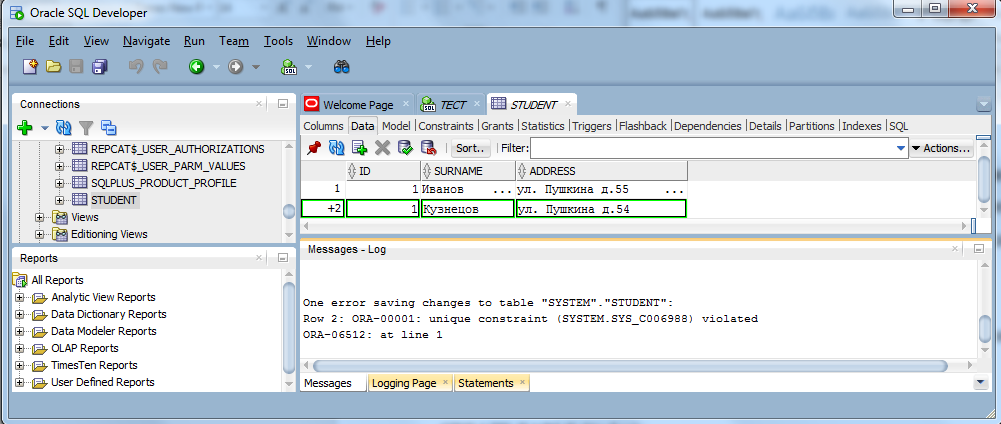
Surname char(30),

Address char(50));

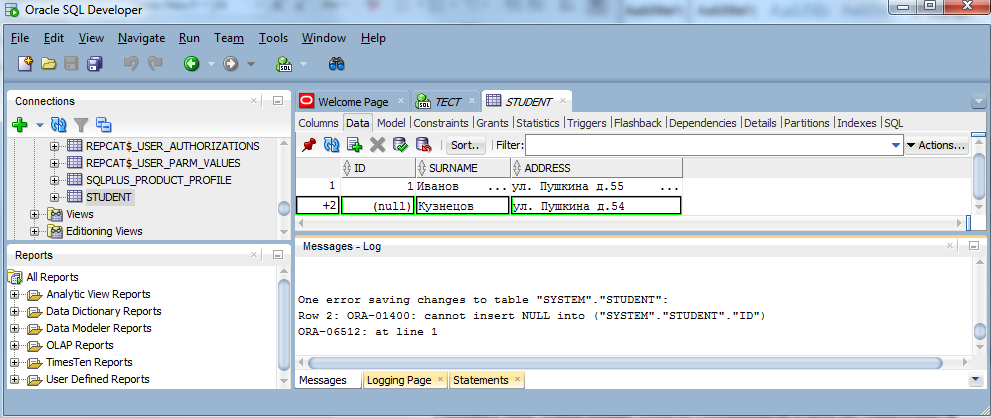
Добавление записи:



Ошибка при добавлении записи (первичный ключ не уникален):



Ошибка при добавлении записи (первичный ключ не может быть пустым):



**Решение 2:**

CREATE TABLE Student

(Id integer NOT NULL,

Surname char(30),

Address char(50));

1.2. Альтернатива для Insert для некоторого организационного правила.

2. Ограничение целостности NOT NULL

Указывает, может ли столбец содержать неопределенные значения. Ограничения NULL и NOT NULL могут задаваться только в синтаксисе *column\_constraint*.

NULL - Указывает, что столбец может содержать неопределенные значения. Если вы не указываете ни NOT NULL ни NULL, то по умолчанию столбец может содержать неопределенные значения.

NOT NULL - Указывает, что столбец не может содержать неопределенных значений. Для удовлетворения этого ограничения любая стока таблицы должна иметь значение этого столбца.

**Задание:**

Создать таблицу студентов с числовым идентификатором, фамилией и адресом.

**Решение 1:**

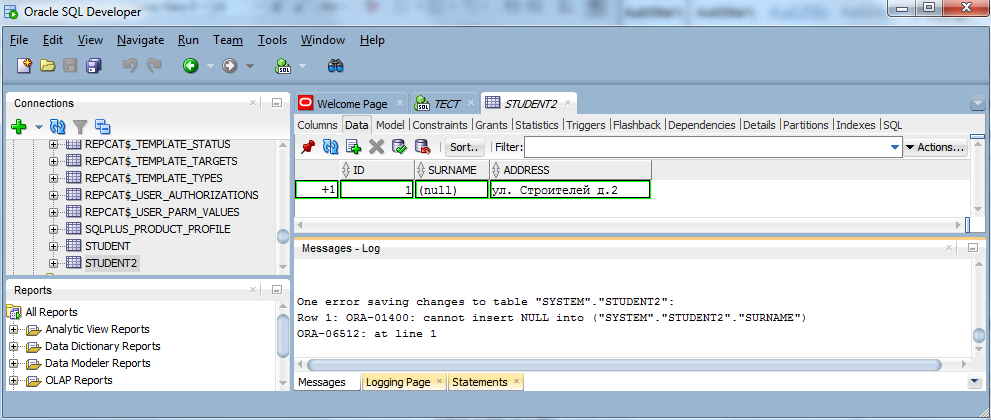
CREATE TABLE Student2

(Id integer NOT NULL PRIMARY KEY,

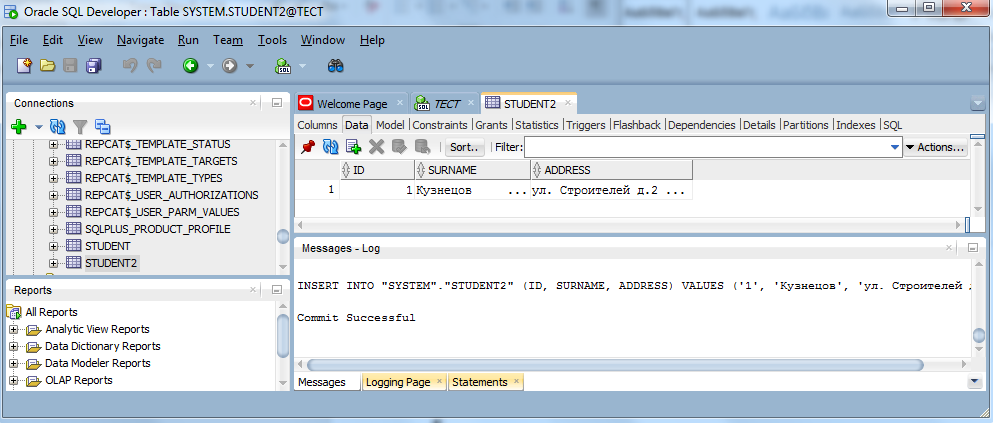
Surname char(30) NOT NULL,

Address char(50));

Ошибка при добавлении записи (поле не может быть пустым):



Добавление записи:



**Решение 2:**

CREATE TABLE Student2

(Id integer NOT NULL,

Surname char(30) NOT NULL,

Address char(50));

3. Ограничений целостности UNIQUE

Указывает, что столбец или совокупность столбцов должны удовлетворять ограничению UNIQUE. Ограничение UNIQUE указывает, что ни какие две строки таблицы не могут содержать одинаковых значений столбца(ов) с этим ограничением. Однако уникальный ключ, состоящий из единственного столбца, может содержать NULL значения.

**Составной уникальный ключ** состоит из совокупности столбцов. Для определения составного уникального ключа необходимо воспользоваться синтаксисом *table\_constraint*, а не синтаксисом *column\_constraint*. Считается, что строка, содержащая NULL значения во всех столбцах составного уникального ключа, удовлетворяет этому ограничению. Однако строки, которые имеют NULL значения для одних из столбцов составного уникального ключа, и одинаковые совокупности значений для оставшихся столбцов, нарушают это ограничение.

Вы не можете использовать один и тот же столбец (совокупность столбцов) в качестве уникального ключа и первичного ключа одновременно.

**Задание:**

Создать таблицу студентов с числовым идентификатором, фамилией и адресом.

**Решение 1:**

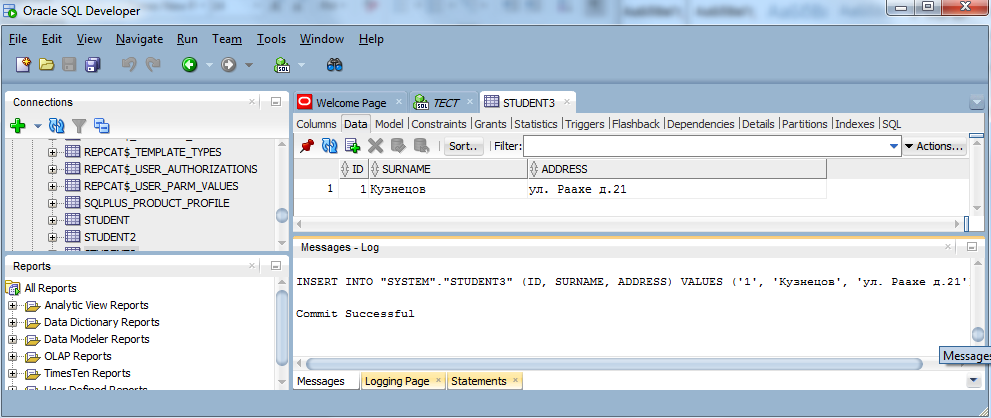
CREATE TABLE Student3

(Id integer NOT NULL UNIQUE,

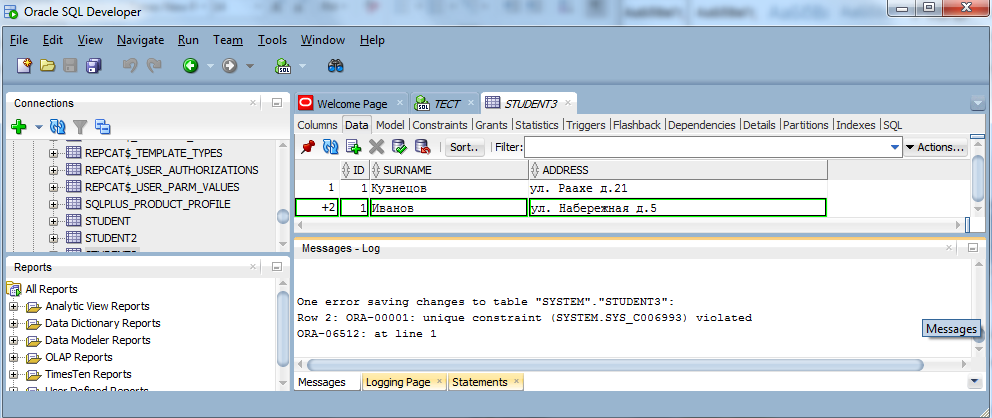
Surname char(30) UNIQUE,

Address char(50));

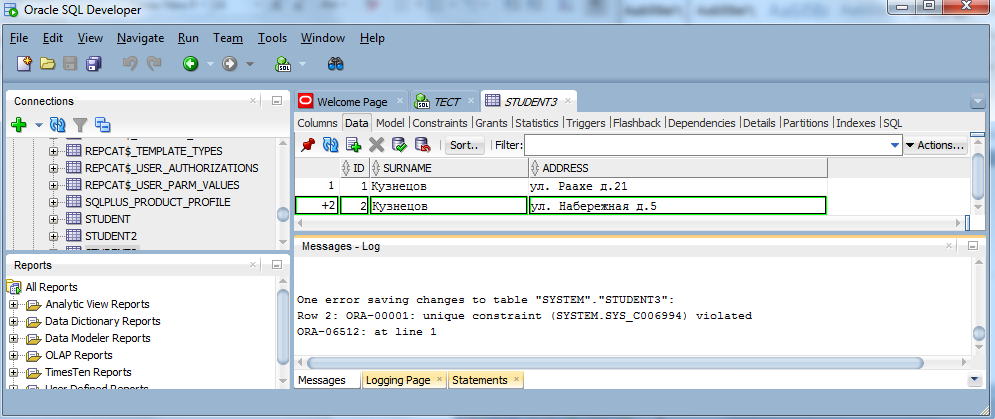
Добавление записи:



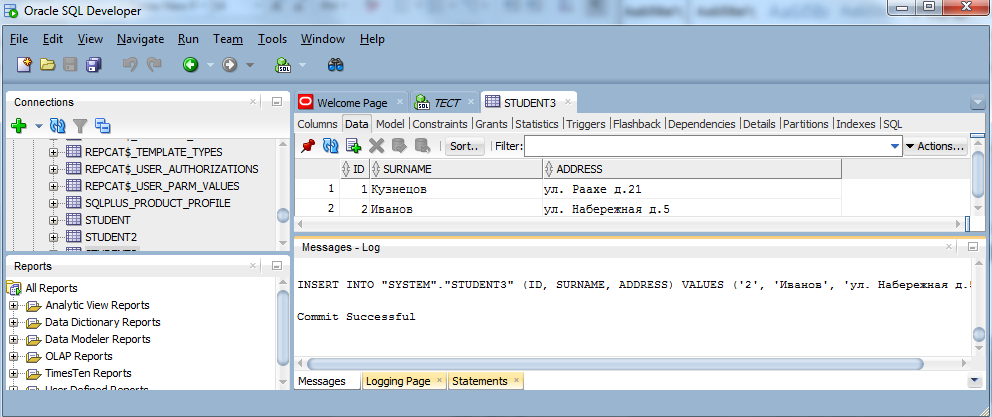
Ошибка при добавлении записи (поле ID не уникально):



Ошибка при добавлении записи (поле Surname не уникально):



Добавление записи с уникальными идентификатором и фамилией (ID и Surname):



**Решение 2:**

CREATE TABLE Student3

(Id integer NOT NULL,

Surname char(30) NOT NULL,

Address char(50));

4. Ссылочные ограничения целостности

Референциальное ограничение целостности указывает, что столбец или совокупность столбцов выступают в качестве внешнего ключа и устанавливает связь между этим внешним ключом и конкретным первичным или уникальным ключом, который называется **ключом, на который делается ссылка**. Таблица, содержащая внешний ключ, называется дочерней, а таблица, на которую делается ссылка, называется родительской. Внешний ключ и ключ, на который делается ссылка, могут быть в одной таблице. В этом случае родительская и дочерние таблицы выступают в одном лице.

Для спецификации референциального ограничения целостности на уровне таблицы используйте фразу *foreign\_key\_clause* в синтаксисе *table\_constraint*. В этом синтаксисе можно определить **составной внешний ключ**, который состоит из совокупности столбцов.

Для спецификации референциального ограничения целостности на уровне столбца используйте фразу REFERENCES синтаксиса *column\_constraint*. В этом случае внешний ключ состоит из единственного столбца

Один и тот же столбец (совокупность столбцов) может выступать в качестве внешнего ключа, и первичного или уникального ключа.

Таблица может содержать множество внешних ключей. Один и тот же столбец может входить в состав многих внешних ключей.

ПО СУТИ РЕФЕРЕНЦИАЛЬНОЕ ОГРАНИЧЕНИЕ ЦЕЛОСТНОТИ ОЗНАЧАЕТ, ЧТО ВНЕШНИЙ КЛЮЧ НЕ МОЖЕТ ССЫЛАТЬСЯ НА ОТСУТСТВУЮЩЕЕ ЗНАЧЕНИЕ ТОГО КЛЮЧА, НА КОТОРЫЙ ДЕЛАЕТСЯ ССЫЛКА

**Ограничения:**

* Ключ UNIQUE или PRIMARY , на который делается ссылка, должен уже быть определен.
* Родительская и дочерняя таблицы должны быть в одной базе данных.
* Нельзя определять референциальное ограничение целостности в предложении CREATE TABLE, которое содержит фразу AS *subquery*. В этом случае сначала создайте таблицу без ограничения, а затем добавьте его позже с помощью предложения ALTER TABLE.

Синтаксис SQL для определения ссылочной целостности выглядит, по существу, подобно следующему. Слова из прописных буква обозначают ключевые слова. Квадратные скобки выделяют необязательные параметры. Столбцы внешнего ключа находятся в таблице *table1*, а столбцы первичного ключа (или другой комбинации столбцов со свойством уникальности значений) – в таблице *table2*.

ALTER TABLE tableName1

ADD CONSTRAINT constraintName

FOREIGN KEY (columnList)

REFERENCES tableName2 [(columnList)]

[onDeleteAction] [onUpdateAction];

**Задание:**

Создать таблицу студентов с числовым идентификатором, которая ссылается на таблицу с адресами.

**Решение 1:**

Столбцы внешнего ключа находятся в таблице table1, а столбцы первичного ключа (или другой комбинации столбцов со свойством уникальности значений) – в таблице table2.

CREATE TABLE StudentsTable

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_id integer);

CREATE TABLE AddressTable

(Id integer NOT NULL PRIMARY KEY,

Address char(50));

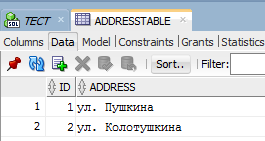
ALTER TABLE StudentsTable

ADD CONSTRAINT StudentsAndAddress

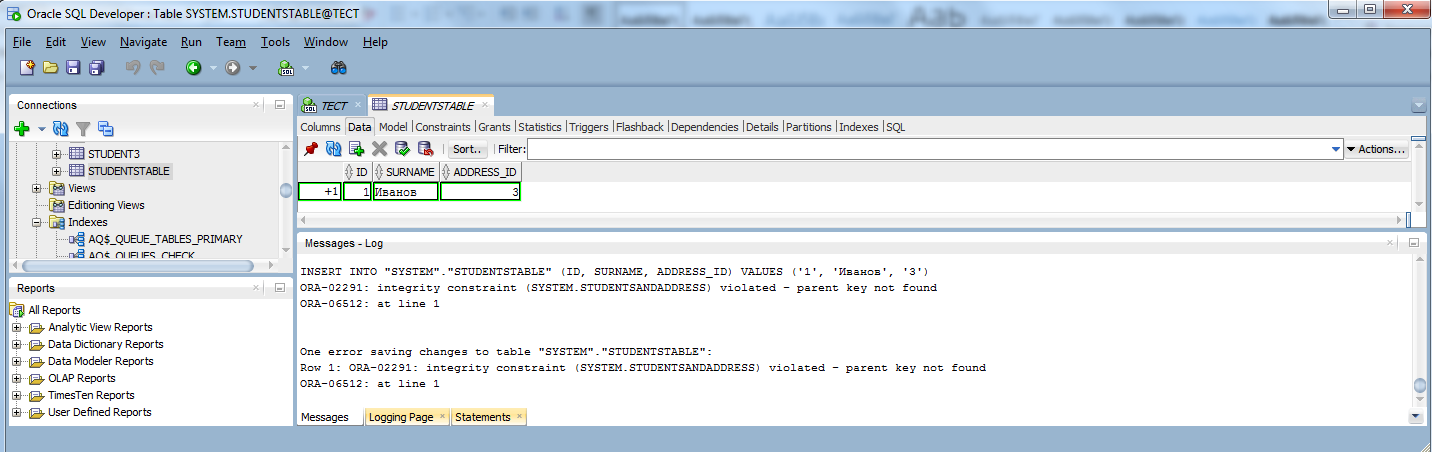
FOREIGN KEY (Address\_id)

REFERENCES AddressTable;

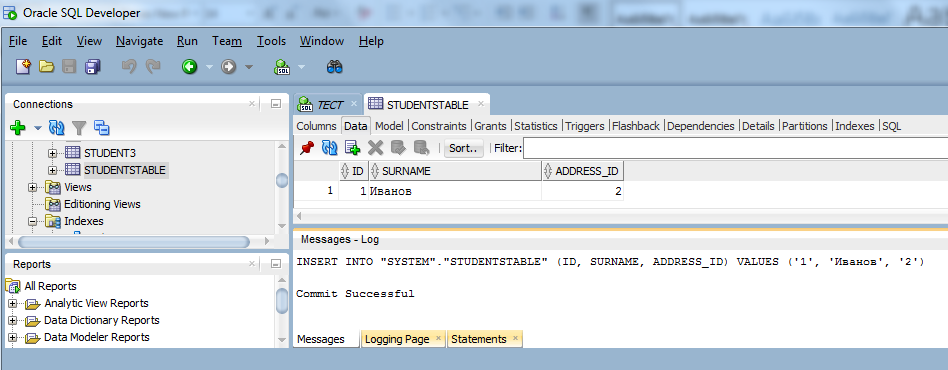
Таблица с адресами:



Ошибка при добавлении записи (идентификатор отсутствует в таблице адресов):



Добавление записи:



**Решение 2:**

CREATE TABLE StudentsTable

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_id integer);

CREATE TABLE AddressTable

(Id integer NOT NULL PRIMARY KEY,

Address char(50));

5. Связи между родительскими и порожденными таблицами

При использовании ALTER TABLE вместо CREATE TABLE, для применения ограничения **FOREIGN KEY**, значения, указываемые во внешнем ключе и родительском ключе, должны быть в состоянии ссылочной целостности. Иначе команда будет отклонена.

Используя ограничение **FOREIGN KEY** таблицы или столбца, можно не указывать список столбцов родительского ключа, если родительский ключ имеет ограничение **PRIMARY KEY**. Естественно, в случае ключей со многими полями, порядок столбцов во внешних и первичных ключах должен совпадать, и, в любом случае, принцип совместимости между двумя ключами все еще применим

Поддержание ссылочной целостности требует некоторых ограничений на значения, которые  могут быть представлены в полях,  объявленных как внешний ключ и родительский ключ. Родительский ключ должен быть структурен, чтобы гарантировать, что каждое значение внешнего ключа будет соответствовать одной указанной строке. Это означает, что он (ключ) должен быть уникальным и не содержать никаких пустых значений(NULL).

Этого не достаточно для родительского ключа в случае выполнения такого требования,  как при объявлении внешнего ключа. **SQL** должен быть уверен, что двойные значения или пустые значения (NULL) не были введены в  родительский ключ.  Следовательно необходимо убедиться, что все поля, которые используются как родительские ключи,  имеют или ограничение PRIMARY **KEY** или ограничение UNIQUE, наподобие ограничения NOT NULL.

Ссылка внешних  ключей только на первичные ключи — хорошая стратегия. Когда используются внешние  ключи,  они связываются не просто с родительскими ключами, на которые они ссылаются; они связываются с определенной строкой таблицы, где этот родительский ключ будет найден. Сам по себе родительский ключ не обеспечивает никакой информации, которая бы не была уже представлена во внешнем ключе.

Так как цель первичного ключа состоит в том,  чтобы идентифицировать уникальность строки, это более логичный и менее  неоднозначный  выбор для внешнего ключа.  Для любого внешнего ключа, который использует уникальный ключ как родительский ключ, необходимо создать внешний ключ, который бы использовал первичный ключ той же самой таблицы для того же самого действия. Внешний ключ, который не имеет никакой другой цели кроме связывания строк, напоминает первичный ключ, используемый исключительно для идентификации строк, и является хорошим средством сохранения структуры  базы данных ясной и простой. Внешний ключ может содержать только те значения, которые фактически представлены в родительском ключе или пустые (NULL). Попытка ввести другие значения в этот ключ будет отклонена.

5.1. ВНЕШНИЙ КЛЮЧ БЕЗ ОГРАНИЧЕНИЙ.

**Задание:**

Создать таблицу студентов с числовым идентификатором, которая ссылается на таблицу с адресами.

1 Иванов 2

1 ул. Пушкина

2 ул. Колотушкина

**Решение 1:**

CREATE TABLE StudentsTable1

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_id integer);

CREATE TABLE AddressTable1

(Id integer,

Address char(50));

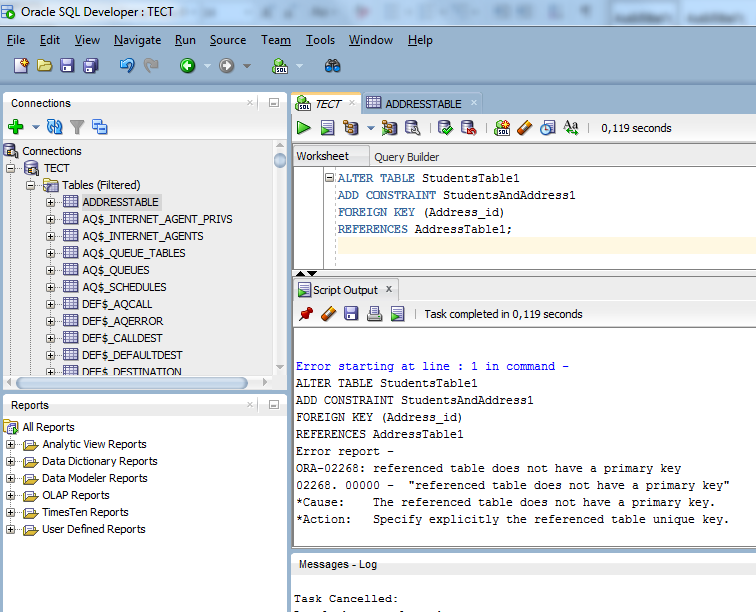
ALTER TABLE StudentsTable1

ADD CONSTRAINT StudentsAndAddress1

FOREIGN KEY (Address\_id)

REFERENCES AddressTable1;

Создание ссылки на таблицу без ключевого поля:



**Решение 2:**

CREATE TABLE StudentsTable1

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_house integer);

CREATE TABLE AddressTable1

(Id integer PRIMARY KEY,

Street char(50),

House integer);

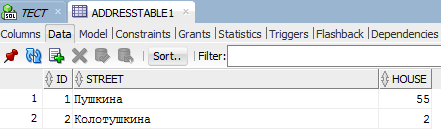
ALTER TABLE StudentsTable1

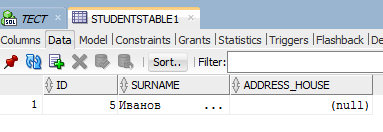
ADD CONSTRAINT StudentsAndAddress1

FOREIGN KEY (Address\_house)

REFERENCES AddressTable1;

Результат:





5.2. ОГРАНИЧЕНИЕ NOT NULL ПО ВНЕШНЕМУ КЛЮЧУ.

**Решение:**

CREATE TABLE StudentsTable2

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_house integer);

CREATE TABLE AddressTable2

(Id integer PRIMARY KEY,

Street char(50),

House integer NOT NULL);

ALTER TABLE StudentsTable2

ADD CONSTRAINT StudentsAndAddress2

FOREIGN KEY (Address\_house)

REFERENCES AddressTable2;

5.3. ОГРАНИЧЕНИЕ UNIQUE ПО ВНЕШНЕМУ КЛЮЧУ.

**Решение:**

CREATE TABLE StudentsTable2

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_house integer);

CREATE TABLE AddressTable2

(Id integer PRIMARY KEY,

Street char(50),

House integer UNIQUE);

ALTER TABLE StudentsTable2

ADD CONSTRAINT StudentsAndAddress2

FOREIGN KEY (Address\_house)

REFERENCES AddressTable2;

5.4. ОГРАНИЧЕНИЯ UNIQUE И NOT NULL ПО ВНЕШНЕМУ КЛЮЧУ.

**Решение:**

CREATE TABLE StudentsTable2

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Address\_house integer);

CREATE TABLE AddressTable2

(Id integer PRIMARY KEY,

Street char(50),

House integer NOT NULL UNIQUE);

ALTER TABLE StudentsTable2

ADD CONSTRAINT StudentsAndAddress2

FOREIGN KEY (Address\_house)

REFERENCES AddressTable2;

6. Ограничение целостности CHECK

Специфицирует условие, которому должна удовлетворять любая строка таблицы. Строка удовлетворяет условию, если на ней это условие принимает значение TRUE или UNKNOWN. Когда Oracle вычисляет условие ограничения CHECK относительно конкретной строки, то имена столбцов в выражении условия ссылаются на значения этих столбцов в этой строке.

При определении для столбца многих ограничений CHECK стройте их так, чтобы не было конфликтов между ними. Oracle не проверяет условия в CHECK на их взаимную противоречивость.

Условие в ограничении CHECK может ссылаться на любой столбец таблицы, но не может ссылаться на столбцы другой таблицы.

**Задача:** Создать таблицу преподавателей с их зарплатой.

6.1. Предложите такую задачу, которая могла быть решена:

а) с применением ограничения целостности CHECK

**Решение:**

CREATE TABLE Teachers

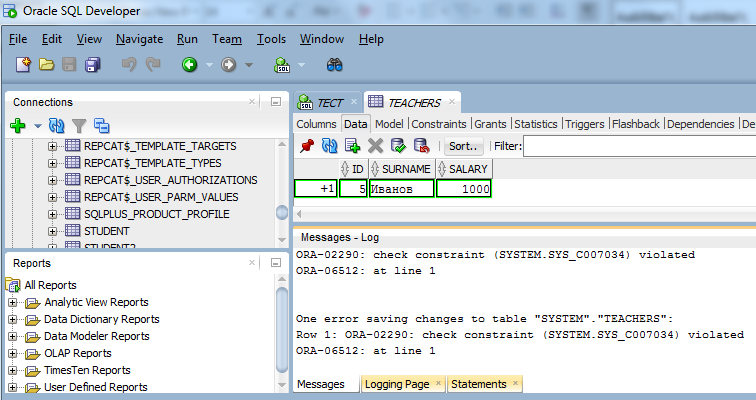
(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

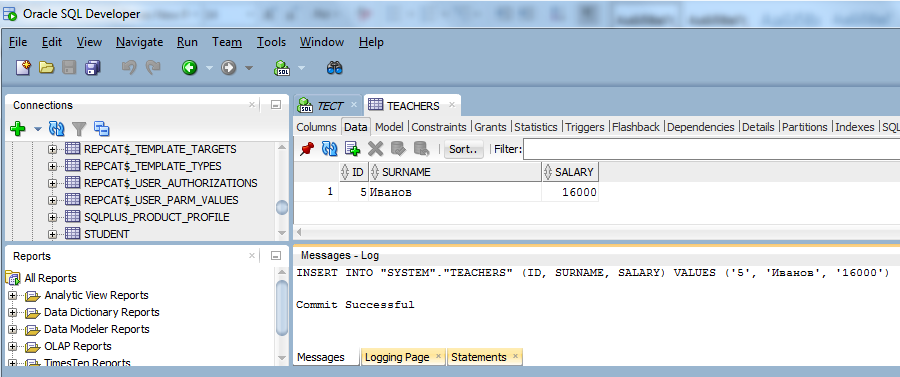
Salary integer

CONSTRAINT MaxAndMinSalary CHECK (Salary BETWEEN 15000 and 35000));

Ошибка при добавлении записи (Salary меньше заданного значения):



Добавление записи:



б) без применения ограничения целостности CHECK

**Решение:**

CREATE TABLE Teachers

(Id integer NOT NULL PRIMARY KEY,

Surname char(30) NOT NULL,

Salary integer);

6.2. Предложите такую задачу, в которой ограничение целостности CHECK имеет:

а) истинное условие;

б) ложное условие.

7. Ограничение целостности CONSTRAINT

Дает возможность идентифицировать (именовать) ограничение с помощью имени *constraint*.  Если вы не указываете ни ограничение NULL ни ограничение NOT NULL в определении столбца, то по умолчанию предполагается NULL. Эта конструкция является факультативной. При ее отсутствии Oracle самостоятельно присваивает имя ограничению целостности.

**Задача:** Создать таблицу преподавателей с уникальным id.

**Решение 1:**

CREATE TABLE Teachers2

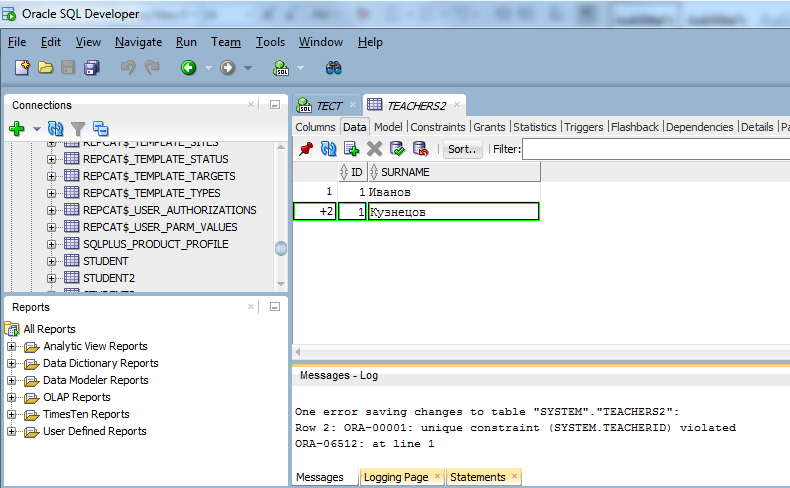
(Id integer NOT NULL,

Surname char(30) NOT NULL,

CONSTRAINT TeacherId UNIQUE(Id)

);

Ошибка при добавлении записи с имеющимся id:



**Решение 2:**

CREATE TABLE Teachers2

(Id integer NOT NULL,

Surname char(30) NOT NULL);

8. Проблемы команды ALTER TABLE

1. Потеря данных.

2.Изменение структуры больших таблиц занимает очень много времени;

9. Выключение ограничений целостности

а) для двух любых ранее рассмотренных задач реализуйте ВЫКЛючение ограничения целостности;

**Решение:**

alter table Teachers2 disable constraint TeacherId

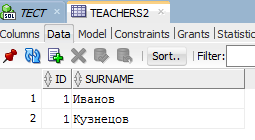
alter table Teachers disable constraint MaxAndMinSalary

б) измените данные в таблице с нарушением этого ограничения; ВКЛючите ограничение целостности и оцените реакцию на эту проблемную ситуацию.

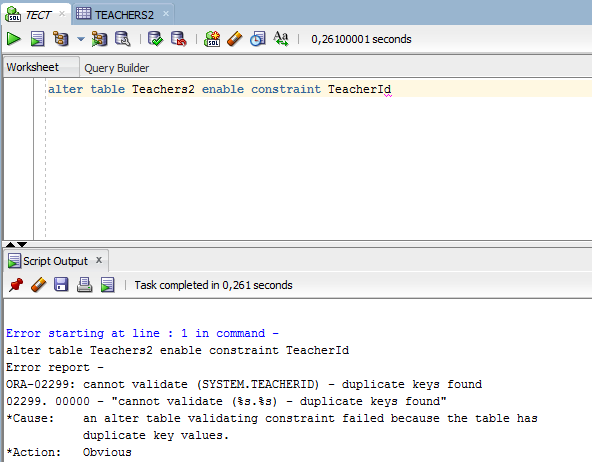
**Решение:**

alter table Teachers2 disable constraint TeacherId

Изменение таблицы (добавление нарушения ограничения):



Попытка включить ограничение:



10. Изменение ограничений целостности

Попытайтесь изменить существующее ограничение целостности; оцените реакцию.

Нельзя изменять ограничения когда-либо, но их можно отбросить, а затем воссоздать.

Пример:

ALTER TABLE your\_table DROP CONSTRAINT ACTIVEPROG\_FKEY1;

ALTER TABLE your\_table

add CONSTRAINT ACTIVEPROG\_FKEY1 FOREIGN KEY(ActiveProgCode) REFERENCES PROGRAM(ActiveProgCode)

ON DELETE CASCADE;

11. Удаление ограничений целостности

ALTER TABLE Teachers2

DROP CONSTRAINT TeacherId;