

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет» РТУ МИРЭА

Институт кибербезопасности и цифровых технологий

Кафедра КБ-1 «Защита информации»

ОТЧЁТ

# по практическому занятию №6

**по дисциплине «Безопасность систем баз данных»**

Студент: Кутьин З.С.

Шифр учебной группы:

БББО-05-20

Принял: Кунин Н.Т.

**Задание 1.**

**Дано:**

CREATE TABLE company (

id INT2 NOT NULL,

short\_name VARCHAR(50) NOT NULL,

full\_name VARCHAR(250) NOT NULL,

direction VARCHAR(50) NOT NULL,

city VARCHAR(150),

phone VARCHAR(15));

Выборка из базы данных предполагается в Министерстве (полный доступ к истинным данным), на фондовой бирже (необходимо подменить данные о направлении деятельности компании – direction на маскирующие), в интернете (необходимо подменить данные о направлении деятельности – direction, полном наименовании – full\_name и телефоне компании – phone). Атрибут id не показывать нигде!

Разработать механизм защиты данных на основании многозначности. Открыть доступ к данным со стороны фондовой биржи и интернета только для чтения.

**Решение:**

* Создание роли для министерства:

CREATE ROLE ministery;

GRANT SELECT ON min\_company TO ministery;

GRANT INSERT ON company TO ministery;

GRANT UPDATE ON company TO ministery;

GRANT DELETE ON company TO ministery; GO;

* Создание представления для уровня министерства:

CREATE ROLE ministery;

GRANT SELECT ON min\_company TO ministery;

GRANT INSERT ON company TO ministery;

GRANT UPDATE ON company TO ministery;

GRANT DELETE ON company TO ministery;

GO;

* Создание таблицы для уровня фондовой биржи и интернета с ненастоящими данными:

CREATE TABLE fake\_list (

fake\_full\_name VARCHAR(250),

fake\_direction VARCHAR(50),

fake\_phone VARCHAR(15)

);

INSERT INTO fake\_list VALUES ('Фейк. название 1', 'Фейк. направление 1', '+79110005678');

INSERT INTO fake\_list VALUES ('Фейк. название 2', 'Фейк. направление 2', '+79110005678');

INSERT INTO fake\_list VALUES ('Фейк. название 3', 'Фейк. направление 3', '+79110005678');

* Внесение данных в таблицу company:

ALTER TABLE company ADD masked\_full\_name VARCHAR(250) NULL;

ALTER TABLE company ADD masked\_direction VARCHAR(50) NULL;

ALTER TABLE company ADD masked\_phone VARCHAR(15) NULL;

GO; UPDATE company

SET masked\_direction = (SELECT TOP 1 fake\_direction FROM fake\_list ORDER BY NEWID())

UPDATE company

SET masked\_full\_name = (SELECT TOP 1 fake\_full\_name FROM fake\_list ORDER BY NEWID())

UPDATE company

SET masked\_phone = (SELECT TOP 1 fake\_phone FROM fake\_list ORDER BY NEWID())

GO;

* Создание представления, в котором столбец direction будут подменен столбцом masked\_direction:

CREATE VIEW fond\_company

AS

SELECT short\_name, full\_name, masked\_direction AS direction, city, phone

FROM company;

GO;

CREATE ROLE fond\_birzha;

GRANT SELECT ON fond\_company TO fond\_birzha;

* Создание представления для интернета:

CREATE VIEW int\_company

AS

SELECT short\_name, masked\_full\_name AS full\_name, masked\_direction AS direction, city,

masked\_phone AS phone

FROM company;

GO;

CREATE ROLE internet;

GRANT SELECT ON int\_company TO internet;

GO;

**Задание 2.**

**Дано:** Для предприятия по продаже канцелярских товаров разработать подсистему сокрытия показателей его экономической деятельности для конкурентов (заниженная прибыль и объем сделок) и потенциальных клиентов (завышенные прибыль, объем сделок и ассортимент товаров). При этом для самого предприятия и налоговой инспекции должны предоставляться истинные данные об основных показателях экономической деятельности. База данных предприятия описывается DDL-скриптом вида:

CREATE TABLE department (id\_dep INT2 NOT NULL,

dep\_name VARCHAR(200) NOT NULL,

fact\_address varchar(350),

phone varchar(12) not null);

CREATE TABLE econ\_indicator (id\_ind INTEGER NOT NULL,

id\_name INT2 NOT NULL,

date\_check DATE NOT NULL,

ind\_val FLOAT);

CREATE TABLE ind\_property (

id\_name INT2 NOT NULL,

indicator\_name VARCHAR(100) NOT NULL,

ind\_note VARCHAR(500));

ALTER TABLE department ADD PRIMARY KEY (id\_dep, id\_dir);

ALTER TABLE ind\_property ADD PRIMARY KEY (id\_name);

ALTER TABLE econ\_indicator ADD

PRIMARY KEY (id\_ind, id\_name, id\_dep);

ALTER TABLE econ\_indicator ADD

FOREIGN KEY (id\_dept) REFERENCES department;

ALTER TABLE econ\_indicator ADD

FOREIGN KEY (id\_name) REFERENCES ind\_property;

**Решение:**

* Для уровня предприятия и налоговой инспекции:

CREATE VIEW department\_info AS

SELECT \*

FROM department;

GO;

CREATE VIEW econ\_indicator\_info AS

SELECT \*

FROM econ\_indicator;

GO;

CREATE VIEW ind\_property\_info AS

SELECT \*

FROM ind\_property;

GO;

CREATE ROLE factory;

GRANT SELECT ON department\_info TO factory;

GRANT SELECT ON econ\_indicator\_info TO factory;

GRANT SELECT ON ind\_property\_info TO factory;

GO;

* Для уровня конкурентов:

CREATE VIEW department\_info\_kon AS

SELECT \*

FROM department;

GO;

CREATE VIEW econ\_indicator\_info\_kon AS

-- Занижение прибыли

SELECT id\_ind, id\_name, date\_check, ind\_val \* 0.75 AS

ind\_val, id\_dep

FROM econ\_indicator;

GO;

CREATE VIEW ind\_property\_info\_kon AS

SELECT \*

FROM ind\_property;

GO;

CREATE ROLE concurents;

GRANT SELECT ON department\_info\_kon TO concurents;

GRANT SELECT ON econ\_indicator\_info\_kon TO concurents;

GRANT SELECT ON ind\_property\_info\_kon TO concurents;

GO;

* Для уровня потенциальных покупателей:

CREATE VIEW department\_info\_cust AS

SELECT \*

FROM department;

GO;

CREATE VIEW econ\_indicator\_info\_cust AS

-- Завышение приыбли

SELECT id\_ind, id\_name, date\_check, ind\_val \* 1.25 AS

ind\_val, id\_dep

FROM econ\_indicator;

GO;

CREATE VIEW ind\_property\_info\_cust AS

SELECT \*

FROM ind\_property;

GO;

CREATE ROLE customers;

GRANT SELECT ON department\_info\_cust TO customers;

GRANT SELECT ON econ\_indicator\_info\_cust TO customers;

GRANT SELECT ON ind\_property\_info\_cust TO customers;

GO;

**Задание 3.**

**Дано:** Запрос в соответствие с SQL-предложением

SELECT DISTINCT dummy FROM ext\_econ\_result;

привел к выдаче вида:

dummy

----------

10

209

228

Сколько уровней благонадежности подсистеме защиты данных?

**Решение:**

* Поскольку выдача SELECT DISTINCT содержит 3 значения, то уровней благонадёжности также 3
* Ответ: 3 уровня

**Задание 4.**

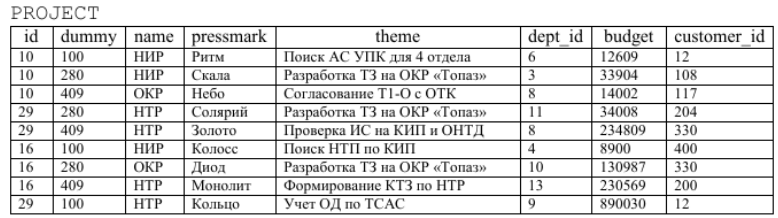
**Дано:** Сколько конфиденциальных данных об 135682 экземплярах объекта учета, если к ней предполагается допустить благонадежности при применении для построения механизма защиты концепции многозначности?

**Решение:**

* Количество записей можно определить по формуле:
* Количество записей = Количество экземпляров ∗ Количество уровней = 135682 ∗ 6 = 814092
* Ответ: 814092 записи

**Задание 5.**

**Дано:**

****

Сформировать SQL-предложения для создания представлений в интересах отделов ОКР (dummy = 409), ФЭО (dummy = 280) и НИО (dummy = 100).

**Решение:**

* Уровень отделов ОКР

CREATE VIEW okr AS

SELECT \*

FROM project

WHERE dummy = 409;

GO;

* Уровень отделов ФЭО

CREATE VIEW feo AS

SELECT \*

FROM project

WHERE dummy = 280;

GO;

* Уровень отделов НИО

CREATE VIEW nio AS

SELECT \*

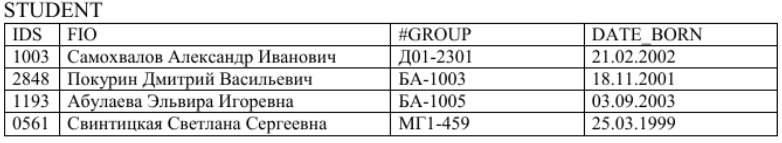
FROM project

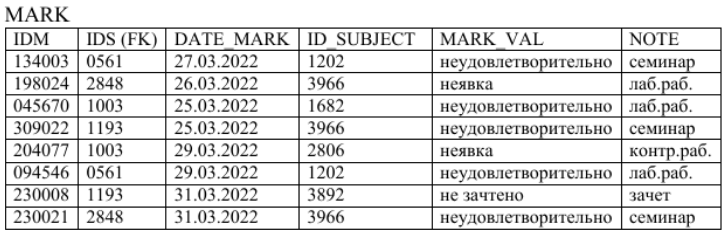
WHERE dummy = 100;

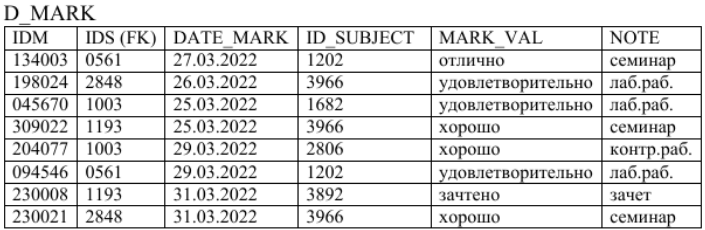
GO;

**Задание 6.**

**Дано:** Разработать хранимую процедуру, которая, в зависимости от параметра DUMMY (0 – истинные данные, 1 – маскирующие данные) формирует выдачу со списком студентов и их оценок из базы данных:







При этом по параметру DUMMY = 0 выводятся данные из таблицы MARK, а по параметру DUMMY = 1 – из таблицы D\_MARK.

**Решение:**

* Разработанная процедура создаёт два курсора: первый проходит по таблице STUDENT, второй — по таблице MARK или D\_MARK (в зависимости от dummy).
* Информация выводится на экран с помощью PRINT:

CREATE OR ALTER PROCEDURE get\_data (@dummy INTEGER)

AS

DECLARE @ids1 INTEGER;

DECLARE @fio1 VARCHAR(30);

DECLARE @group1 VARCHAR(10);

DECLARE @date\_born1 DATE;

DECLARE @idm2 INTEGER;

DECLARE @ids2 INTEGER;

DECLARE @date\_mark2 VARCHAR(15);

DECLARE @id\_subject2 INTEGER;

DECLARE @mark\_val2 VARCHAR(20);

DECLARE @note2 VARCHAR(20);

BEGIN

DECLARE Cur1 CURSOR FOR

SELECT \*

FROM STUDENT;

IF @dummy = 0

BEGIN

DECLARE Cur2 CURSOR FOR

SELECT IDM, DATE\_MARK, ID\_SUBJECT, MARK\_VAL, NOTE

FROM MARK;

END;

ELSE

BEGIN

DECLARE Cur2 CURSOR FOR

SELECT IDM, DATE\_MARK, ID\_SUBJECT, MARK\_VAL, NOTE

FROM D\_MARK;

END;

OPEN Cur1;

FETCH NEXT FROM Cur1 INTO @ids1, @fio1, @group1, @date\_born1;

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'Student: ' + @fio1 + ', group: ' + @group1 + ', born in: ' +

@date\_born1 + ': ';

OPEN Cur2;

FETCH NEXT FROM Cur2 INTO @idm2, @ids2, @date\_mark2, @id\_subject2,

@mark\_val2, @note2;

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @ids2 = @ids1

BEGIN

PRINT ' -> ' + @date\_mark2 + ' ' + @id\_subject2 + ' ' + @mark\_val2

+ ' ' + @note2;

END;

FETCH NEXT FROM Cur2 INTO @idm2, @ids2, @date\_mark2, @id\_subject2,

@mark\_val2, @note2;

END;

CLOSE Cur2;

FETCH NEXT FROM Cur1 INTO @ids1, @fio1, @group1, @date\_born1;

END;

CLOSE Cur2;

DEALLOCATE Cur1;

DEALLOCATE Cur2;

END;