APPENDIX 1. SQL SOURCE CODE FOR THE CONSIDERED EXERCISES

A1.1 Database creation using SQL language

```
-- create database

CREATE DATABASE supply;

-- main database

CREATE TABLE supplier (
supplier_id int NOT NULL,
supplier_address varchar(100) NOT NULL,
supplier_phone varchar(20) NOT NULL,
PRIMARY KEY (supplier_id)
) ENGINE=InnoDB;

CREATE TABLE supplier_person (
supplier_id int NOT NULL,
supplier_last_name varchar(20) NOT NULL,
supplier_first_name varchar(20) NOT NULL,
supplier_middle_name varchar(20) NOT NULL,
PRIMARY KEY (supplier_id),
FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
) ENGINE=InnoDB;

CREATE TABLE supplier_org (
supplier_id int NOT NULL,
supplier_org_name varchar(20) NOT NULL,
PRIMARY KEY (supplier_id),
```

```
FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
) ENGINE=InnoDB;

CREATE TABLE contract (
contract_number int NOT NULL AUTO_INCREMENT,
contract_date timestamp NOT NULL,
supplier_id int NOT NULL,
contract_note varchar(100),
PRIMARY KEY (contract_number),
FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
) ENGINE=InnoDB;

CREATE TABLE supplied (
contract_number int NOT NULL,
supplied_product varchar(20) NOT NULL,
supplied_amount decimal(4,0) NOT NULL,
supplied_cost decimal(8,2) NOT NULL,
PRIMARY KEY (contract_number, supplied_product),
FOREIGN KEY (contract_number) REFERENCES
contract(contract_number)
) ENGINE=InnoDB;

-- additional table

CREATE TABLE contract_delivered (
contract_number int NOT NULL,
delivery_date timestamp NOT NULL,
delivery_note varchar(100),
PRIMARY KEY (contract_number)
) ENGINE=InnoDB;
```

```
-- add relation

ALTER TABLE contract_delivered
ADD CONSTRAINT contract_number_fk FOREIGN KEY (contract_number) REFERENCES contract(contract_number);

-- drop table

DROP TABLE contract_delivered;
```

A1.2 Data manipulation using SQL language: insert, update, and delete

```
-- insert queries

INSERT INTO supplier (supplier_id, supplier_address, supplier_phone) VALUES (1, 'Kharkiv, Nauky av., 55, apt. 108', 'phone: 32-18-44');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone) VALUES (2, 'Kyiv, Peremohy av., 154, apt. 3', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone) VALUES (3, 'Kharkiv, Pushkinska str., 77', 'phone: 33-33-44, fax: 22-12-33');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone) VALUES (4, 'Odesa, Derebasivska str., 75', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone) VALUES (5, 'Poltava, Soborna str., 15, apt. 43', '');

INSERT INTO supplier_person (supplier_id, supplier_last_name, supplier_first_name, supplier_middle_name) VALUES (1, 'Petrov', 'Pavlo', 'Petrovych');
INSERT INTO supplier_person (supplier_id, supplier_last_name, supplier_first_name, supplier_middle_name) VALUES (3, 'Ivanov', 'Illia', 'Illych');
INSERT INTO supplier_person (supplier_id, supplier_last_name, supplier_first_name, supplier_middle_name) VALUES (5, 'Sydorov', 'Serhii', 'Stepanovych');

INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (2, 'Interfruit Ltd.');
INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (4, 'Transservice LLC');
```

```
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-01', 1, 'Order 34 on 30.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-10', 1, 'Invoice 08-78 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-23', 3, 'Order 56 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-24', 2, 'Order 74 on 11.09.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-10-02', 2, 'Invoice 09-12 on 21.09.2018');

INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (1, 'TV', 10, 1300);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (1, 'Audio Player', 25, 700);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (1, 'Video Player', 12, 750);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (2, 'Stereo System', 11, 500);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (2, 'Audio Player', 5, 450);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (2, 'Video Player', 8, 450);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (3, 'TV', 52, 900);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (3, 'Audio Player', 11, 550);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (3, 'Monitor', 85, 550);
```

```
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (4, 'TV', 56, 990);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (4, 'Audio Player', 22, 320);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (4, 'Printer', 41, 350);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (5, 'TV', 14, 860);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (5, 'Audio Player', 33, 580);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount, supplied_cost) VALUES (5, 'Video Player', 17, 850);

-- update/delete queries

UPDATE supplied
SET supplied_cost = supplied_cost * 0.95
WHERE contract_number = 4 AND supplied_product = 'Printer';

DELETE FROM supplied WHERE contract_number = 5;
```

A1.3 Data manipulation using SQL language: select queries and their basic features

```
-- select queries

SELECT supplied.contract_number, supplied.supplied_product, supplier.*, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number
AND supplier.supplier_id = contract.supplier_id AND (contract.contract_number = 1
AND contract.supplier_id = 1);

SELECT contract.contract_number, contract.contract_date, supplied.supplied_product,
    supplied.supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
    INNER JOIN supplied ON contract.contract_number = supplied.contract_number
WHERE contract.contract_date BETWEEN '2018-09-05' AND '2018-09-12' AND
    supplier.supplier_id = 1;

SELECT contract.contract_number, contract.contract_date, supplied.supplied_product,
    supplied.supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id) INNER JOIN
    supplied ON contract.contract_number = supplied.contract_number
```

```
WHERE MONTH(contract.contract_date) = 9 AND
YEAR(contract.contract_date) = 2018;

SELECT contract.contract_number, contract.contract_date,
contract.supplier_id,
    SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`
FROM contract INNER JOIN supplied ON contract.contract_number =
supplied.contract_number
GROUP BY contract.contract_number, contract.contract_date,
contract.supplier_id
ORDER BY contract.contract_number;

SELECT contract.contract_number, contract.contract_date,
contract.supplier_id,
    SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`
FROM contract INNER JOIN supplied ON contract.contract_number =
supplied.contract_number
WHERE contract.contract_number < 4
GROUP BY contract.contract_number, contract.contract_date,
contract.supplier_id
ORDER BY contract.contract_number;

SELECT contract.contract_number, contract.contract_date,
contract.contract_note,
    supplier.*, supplied.supplied_amount
FROM contract, supplied, supplier
WHERE contract.contract_number = supplied.contract_number AND
    contract.supplier_id = supplier.supplier_id AND
    supplied.supplied_amount = (SELECT
MAX(supplied.supplied_amount) FROM supplied);
```

```
SELECT * FROM supplier
WHERE supplier_id NOT IN (SELECT supplier_id FROM supplier);

SELECT supplied_product, AVG(supplied_cost) AS `Average cost`
FROM supplied
GROUP BY supplied_product;

SELECT supplied_product, supplied_amount, supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id =
contract.supplier_id)
    INNER JOIN supplied ON contract.contract_number =
supplied.contract_number
WHERE supplied_cost > (SELECT AVG(supplied_cost) FROM supplied);

SELECT supplied_product, supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id =
contract.supplier_id)
    INNER JOIN supplied ON contract.contract_number =
supplied.contract_number
ORDER BY supplied_cost DESC
LIMIT 1;

SELECT supplier.supplier_id, supplier.supplier_address,
    IFNULL(supplier_org.supplier_org_name,
CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
            SUBSTRING(supplier_person.supplier_first_name, 1, 1), '. ',
            SUBSTRING(supplier_person.supplier_middle_name, 1, 1), '. '))
AS `Supplier`
```

```
  FROM (supplier LEFT JOIN supplier_person ON supplier.supplier_id =
supplier_person.supplier_id)
      LEFT JOIN supplier_org ON supplier.supplier_id =
supplier_org.supplier_id;

  SELECT contract.contract_number, contract.contract_date,
      IFNULL(supplier_org.supplier_org_name,
CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
            SUBSTRING(supplier_person.supplier_first_name, 1, 1), '. ',
            SUBSTRING(supplier_person.supplier_middle_name, 1, 1), '. '))
AS `Supplier`,
      SUM(supplied.supplied_amount) AS `Size`,
      SUM(supplied.supplied_cost * supplied.supplied_amount) AS `Total`
  FROM (((supplier LEFT JOIN supplier_person ON supplier.supplier_id =
supplier_person.supplier_id)
      LEFT JOIN supplier_org ON supplier.supplier_id =
supplier_org.supplier_id)
      INNER JOIN contract ON contract.supplier_id = supplier.supplier_id)
      INNER JOIN supplied ON contract.contract_number =
supplied.contract_number
  GROUP BY supplier.supplier_id, supplier.supplier_address,
      IFNULL(supplier_org.supplier_org_name,
CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
            SUBSTRING(supplier_person.supplier_first_name, 1, 1), '. ',
            SUBSTRING(supplier_person.supplier_middle_name, 1, 1), '. '))
  ORDER BY contract.contract_number;

  SELECT supplied.contract_number, contract.contract_date,
      supplied.supplied_product, supplier.supplier_id
  FROM supplied, contract, supplier
```

```
  WHERE contract.contract_number = supplied.contract_number
      AND supplier.supplier_id = contract.supplier_id AND
contract.supplier_id = 1
  UNION
  SELECT supplied.contract_number, contract.contract_date,
      supplied.supplied_product, supplier.supplier_id
  FROM supplied, contract, supplier
  WHERE contract.contract_number = supplied.contract_number
      AND supplier.supplier_id = contract.supplier_id AND
contract.supplier_id = 2
  ORDER BY supplier_id, contract_number;

  SELECT DISTINCT supplied.supplied_product
  FROM supplied, contract
  WHERE contract.contract_number = supplied.contract_number AND
contract.supplier_id = 1
  UNION
  SELECT DISTINCT supplied.supplied_product
  FROM supplied, contract
  WHERE contract.contract_number = supplied.contract_number AND
contract.supplier_id = 2
  ORDER BY supplied_product;

  SELECT supplied_product, COUNT(supplied_product) AS
`SupplyFrequency`
  FROM supplied
  GROUP BY supplied_product
  HAVING COUNT(supplied_product) > 1
  ORDER BY COUNT(supplied_product) DESC;
```

```
SELECT supplied_product, SUM(IF(MONTH(contract_date) = 1, supplied_amount, 0)) AS `Jan`,
    SUM(IF(MONTH(contract_date) = 2, supplied_amount, 0)) AS `Feb`,
    SUM(IF(MONTH(contract_date) = 3, supplied_amount, 0)) AS `Mar`,
    SUM(IF(MONTH(contract_date) = 4, supplied_amount, 0)) AS `Apr`,
    SUM(IF(MONTH(contract_date) = 5, supplied_amount, 0)) AS `May`,
    SUM(IF(MONTH(contract_date) = 6, supplied_amount, 0)) AS `Jun`,
    SUM(IF(MONTH(contract_date) = 7, supplied_amount, 0)) AS `Jul`,
    SUM(IF(MONTH(contract_date) = 8, supplied_amount, 0)) AS `Aug`,
    SUM(IF(MONTH(contract_date) = 9, supplied_amount, 0)) AS `Sep`,
    SUM(IF(MONTH(contract_date) = 10, supplied_amount, 0)) AS `Oct`,
    SUM(IF(MONTH(contract_date) = 11, supplied_amount, 0)) AS `Nov`,
    SUM(IF(MONTH(contract_date) = 12, supplied_amount, 0)) AS `Dec`
FROM contract, supplied
WHERE contract.contract_number = supplied.contract_number AND YEAR(contract_date) = 2018
GROUP BY supplied_product
ORDER BY supplied_product;

SELECT supplied.contract_number, supplied.supplied_product,
    supplied.supplied_amount, supplied.supplied_cost,
    contract.contract_date,
    MONTHNAME(contract.contract_date) AS `Month`,
    YEAR(contract.contract_date) AS `Year`
FROM supplied, contract
WHERE contract.contract_number = supplied.contract_number;
```

A1.4 Creation and usage of views

```
-- views

SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
    supplier_org.supplier_org_name, supplier_person.supplier_last_name,
    supplier_person.supplier_first_name, supplier_person.supplier_middle_name
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
    LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
    LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;

SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
    IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
    supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS `Supplier`
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
    LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
    LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;

CREATE VIEW supplier_info AS
```

```
SELECT supplier.supplier_id, supplier.supplier_address,
    IFNULL(supplier_org.supplier_org_name,
CONCAT(supplier_person.supplier_last_name, ' ',
    supplier_person.supplier_first_name, ' ',
supplier_person.supplier_middle_name)) AS `Info`
FROM supplier LEFT OUTER JOIN supplier_org ON supplier.supplier_id
= supplier_org.supplier_id
    LEFT OUTER JOIN supplier_person ON supplier.supplier_id =
supplier_person.supplier_id;
```

A1.5 Creation and usage of stored procedures and triggers

```
-- stored procedures

DELIMITER //
CREATE PROCEDURE sp_contract()
BEGIN
    SELECT *
    FROM (contract LEFT JOIN supplier_org ON
        contract.supplier_id = supplier_org.supplier_id)
        LEFT JOIN supplier_person ON
        contract.supplier_id = supplier_person.supplier_id;
END //

DELIMITER //
CREATE PROCEDURE sp_contract_total(IN date_from timestamp,
                                                    IN
date_to timestamp)
BEGIN
    SELECT contract.contract_number, contract.contract_date,
        SUM(supplied.supplied_amount),
SUM(supplied.supplied_amount * supplied.supplied_cost)
    FROM contract LEFT JOIN supplied ON contract.contract_number =
supplied.contract_number
    WHERE contract.contract_date BETWEEN date_from AND date_to
    GROUP BY contract.contract_number, contract.contract_date;
END //

CALL sp_contract_total('2018-09-01', '2018-10-31');

DELIMITER //
```

```
CREATE PROCEDURE sp_contract_ops(IN op CHAR(1), IN c_num INT,
IN c_date TIMESTAMP,
                                                    IN s_id
INT, IN c_note VARCHAR(100))
BEGIN
    IF op = 'i' THEN
        INSERT INTO contract(contract_date, supplier_id,
contract_note)
            VALUES(CURRENT_TIMESTAMP(), s_id, c_note);
    ELSEIF op = 'u' THEN
        UPDATE contract SET contract_date = c_date,
                                                supplier_id = s_id,
                                                contract_note = c_note
        WHERE contract_number = c_num;
    ELSE
        DELETE FROM contract WHERE contract_number = c_num;
    END IF;
END //

CALL sp_contract_ops('i', 0, '2018-12-16', 2, 'contract inserted');

CALL sp_contract_ops('u', 6, '2018-12-31', 2, 'contract updated');

CALL sp_contract_ops('d', 6, '2018-12-31', 0, '');

-- triggers

DELIMITER //
CREATE TRIGGER not_null_date BEFORE INSERT ON contract
FOR EACH ROW
```

```
BEGIN
    IF NEW.contract_date IS NULL THEN
        SET NEW.contract_date = CURRENT_TIMESTAMP();
    END IF;
END //

INSERT INTO contract (supplier_id, contract_note) VALUES (1, '');

DELIMITER //
CREATE TRIGGER check_supplier_org BEFORE INSERT ON supplier_person
FOR EACH ROW
BEGIN
    IF NEW.supplier_id IN (SELECT supplier_id FROM supplier_org) THEN
        SET @message = CONCAT('The person with id ', NEW.supplier_id,
            ' is already stored as the organization!');
        SIGNAL SQLSTATE '45001'
        SET MESSAGE_TEXT = @message;
    END IF;
END //

INSERT INTO supplier_person VALUES (2, 'Makarov', 'Oleg', 'Petrovych');
```

A1.6 Basics of data integrity control mechanisms

```
-- referential constraints NO ACTION

ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE supplier_org
DROP FOREIGN KEY supplier_org_ibfk_1;

ALTER TABLE supplier_org
ADD CONSTRAINT supplier_org_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE supplier_person
DROP FOREIGN KEY supplier_person_ibfk_1;

ALTER TABLE supplier_person
ADD CONSTRAINT supplier_person_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

DELETE FROM supplier WHERE supplier_id = 4;

UPDATE supplier SET supplier_id = 7 WHERE supplier_id = 5;
```

```
-- referential constraints CASCADE

ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE supplier_org
DROP FOREIGN KEY supplier_org_ibfk_1;

ALTER TABLE supplier_org
ADD CONSTRAINT supplier_org_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE supplier_person
DROP FOREIGN KEY supplier_person_ibfk_1;

ALTER TABLE supplier_person
ADD CONSTRAINT supplier_person_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

UPDATE supplier SET supplier_id = 8 WHERE supplier_id = 2;

DELETE FROM supplier WHERE supplier_id = 8;
```

```
-- referential constraints SET NULL

ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
MODIFY supplier_id INT NULL;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE SET NULL ON UPDATE SET NULL;

UPDATE supplier SET supplier_id = 10 WHERE supplier_id = 3;
```

A1.7 Work with transactions

```
-- transactions

SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
    supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number AND supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;

SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO supplied VALUES (1, 'Vacuum cleaner', 22, 390);

SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
    supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number AND supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;

ROLLBACK;

SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
    supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
```

```
WHERE contract.contract_number = supplied.contract_number AND
supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;


SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
    VALUES (6, 'Kyiv, Velyka Vasylkivska st., 55', '');
INSERT INTO contract (contract_date, supplier_id, contract_note)
    VALUES ('2018-12-12', 6, '');
INSERT INTO supplied VALUES (6, 'Vacuum cleaner', 22, 390);
INSERT INTO supplied VALUES (6, 'Coffee machine', 33, 90);

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;


ALTER TABLE contract
```

```
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
UPDATE supplier SET supplier_id = 22 WHERE supplier_id = 6;
UPDATE supplied SET supplied_cost = supplied_cost * 1.1 WHERE contract_number = 8;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied WHERE contract_number = 8;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied WHERE contract_number = 8;


ALTER TABLE supplied
DROP FOREIGN KEY supplied_ibfk_1;
```

```
ALTER TABLE supplied
ADD CONSTRAINT supplied_ibfk_1 FOREIGN KEY (contract_number) REFERENCES contract(contract_number) ON DELETE CASCADE ON UPDATE CASCADE;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM supplier WHERE supplier_id = 22;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

-- isolation levels

#SET GLOBAL TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
#SET GLOBAL TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
#SET GLOBAL TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET GLOBAL TRANSACTION ISOLATION LEVEL SERIALIZABLE;

SET AUTOCOMMIT = 0;
START TRANSACTION;
#INSERT INTO supplier VALUES (23, 'Kharkiv, Kyrpychova st., 2', '');
#UPDATE supplier SET supplier_phone = '223-322' WHERE supplier_id = 22;
SELECT * FROM supplier;
DO SLEEP(10);
SELECT * FROM supplier;
ROLLBACK;
```

A1.8 User rights management

```
-- users

CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin123';

CREATE USER 'manager'@'localhost' IDENTIFIED BY 'manager123';

CREATE USER 'storekeeper'@'localhost' IDENTIFIED BY
'storekeeper123';

SELECT Host, User, Password FROM mysql.user;

GRANT ALL ON supply.* TO 'admin'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON supply.supplier TO
'manager'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON supply.supplier_org TO
'manager'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON supply.supplier_person
TO 'manager'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON supply.contract TO
'manager'@'localhost';

GRANT SELECT ON supply.supplied TO 'manager'@'localhost';

GRANT EXECUTE ON supply.* TO 'manager'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON supply.supplied TO
'storekeeper'@'localhost';

GRANT SELECT ON supply.contract TO 'storekeeper'@'localhost';

GRANT EXECUTE ON supply.* TO 'storekeeper'@'localhost';

SELECT * FROM mysql.db

WHERE Db = 'supply';
```

```
SELECT Db, User, Table_name, Table_priv FROM mysql.tables_priv
WHERE Db = 'supply';
```