3a

```sql
DROP TABLE IF EXISTS Books, Customers, Carts, Orders, Order_items CASCADE;

CREATE TABLE Books (
        isbn            TEXT UNIQUE,
        title           TEXT NOT NULL,
        authors         TEXT NOT NULL,
        year            INTEGER,
        edition         TEXT NOT NULL
                        CHECK (edition in ("paperback", "hardcover", "ebook")),
        publisher       TEXT,
        number_pages    INTEGER
                        CHECK (number_pages == null or number pages > 0),
        prices          NUMERIC
                        CHECK  (prices > 0),
        PRIMARY KEY (isbn)
};

CREATE TABLE Customers (
        cust_id         INTEGER UNIQUE,
        name            TEXT NOT NULL,
        email           TEXT,
        PRIMARY KEY (cust_id)
);

CREATE TABLE Carts (
        cust_id         INTEGER,
        isbn            TEXT,
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id),
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);

CREATE TABLE Orders (
        order_id        INTEGER UNIQUE,
        order_date      DATE,
        cust_id         TEXT,
        PRIMARY KEY (order_id),
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id)
);

CREATE TABLE Order_items (
        order_id        INTEGER,
        isbn            TEXT,
        FOREIGN KEY (order_id) REFERENCES Orders (order_id)
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);
```

3b. Change the data constraint of order_timestamp from DATE to TIMESTAMP. Add UNIQUE constraint to the attribute.

3c 1.

DROP TABLE IF EXISTS Books, Customers, Carts, Orders, Order_items CASCADE;

```
CREATE TABLE Books (
        isbn            TEXT UNIQUE,
        title           TEXT NOT NULL,
        authors         TEXT NOT NULL,
        year            INTEGER,
        edition         TEXT NOT NULL
                        CHECK (edition in ("paperback", "hardcover", "ebook"))
        publisher       TEXT,
        number_pages INTEGER
                        CHECK (number_pages == null or number pages > 0),
        prices          NUMERIC
                        CHECK  ((prices > 0 and edition <> "ebook") or (prices >= 30 and edition ==
                        "ebook"))

        PRIMARY KEY (isbn)
};

CREATE TABLE Customers (
        cust_id         INTEGER UNIQUE,
        name            TEXT NOT NULL,
        email           TEXT,
        PRIMARY KEY (cust_id)
);

CREATE TABLE Carts (
        cust_id         INTEGER,
        isbn            TEXT,
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id),
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);

CREATE TABLE Orders (
        order_id        INTEGER UNIQUE,
        order_date      DATE,
        cust_id         TEXT,
        PRIMARY KEY (order_id),
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id)
);

CREATE TABLE Order_items (
        order_id        INTEGER,
        isbn            TEXT,
```

```
        FOREIGN KEY (order_id) REFERENCES Orders (order_id)
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);
```

3c 2. Cannot be expressed

3C 3.

```sql
DROP TABLE IF EXISTS Books, Customers, Carts, Orders, Order_items CASCADE;

CREATE TABLE Books (
        isbn            TEXT UNIQUE,
        title           TEXT NOT NULL,
        authors         TEXT NOT NULL,
        year            INTEGER,
        edition         TEXT NOT NULL
                        CHECK (edition in ("paperback", "hardcover", "ebook")),
        publisher       TEXT,
        number_pages    INTEGER
                        CHECK (number_pages == null or number pages > 0),
        prices          NUMERIC
                        CHECK  ((prices > 0 and edition <> "ebook") or (prices >= 30 and edition ==
                        "ebook"))

        PRIMARY KEY (isbn)
};

CREATE TABLE Customers (
        cust_id         INTEGER UNIQUE,
        name            TEXT NOT NULL,
        email           TEXT,
        PRIMARY KEY (cust_id)
);

CREATE TABLE Carts (
        cust_id         INTEGER,
        isbn            TEXT,
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id),
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);

CREATE TABLE Orders (
        order_id        INTEGER UNIQUE,
        order_date      DATE,
        cust_id         TEXT,
        PRIMARY KEY (order_id),
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id)
);

CREATE TABLE Order_items (
        order_id        INTEGER,
        isbn            TEXT,
        FOREIGN KEY (order_id) REFERENCES Orders (order_id)
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);
```

3C 4.

```sql
DROP TABLE IF EXISTS Books, Customers, Carts, Orders, Order_items CASCADE;

CREATE TABLE Books (
        isbn            TEXT UNIQUE,
        title           TEXT NOT NULL,
        authors         TEXT NOT NULL,
        year            INTEGER,
        edition         TEXT NOT NULL
                        CHECK (edition in ("paperback", "hardcover", "ebook")),
        publisher       TEXT,
                        CHECK (publisher <> "ACME" or (publisher == "ACME" and year >= 2010 and
                        edition == "ebook")),
        number_pages    INTEGER
                        CHECK (number_pages == null or number pages > 0),
        prices          NUMERIC
                        CHECK  (prices > 0),
        PRIMARY KEY (isbn)
};

CREATE TABLE Customers (
        cust_id         INTEGER UNIQUE,
        name            TEXT NOT NULL,
        email           TEXT,
        PRIMARY KEY (cust_id)
);

CREATE TABLE Carts (
        cust_id         INTEGER,
        isbn            TEXT,
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id),
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);

CREATE TABLE Orders (
        order_id        INTEGER UNIQUE,
        order_date      DATE,
        cust_id         TEXT,
        PRIMARY KEY (order_id),
        FOREIGN KEY (cust_id) REFERENCES Customers (cust_id)
);

CREATE TABLE Order_items (
        order_id        INTEGER,
        isbn            TEXT,
        FOREIGN KEY (order_id) REFERENCES Orders (order_id)
```

```
        FOREIGN KEY (isbn) REFERENCES Books (isbn)
);
```