PRINCIPLES OF DATABASE SYSTEMS

PROJECT PART 1

FA22_CS-GY 6083-B

Prof. Amit Patel

ADITYA NARANG (an3731)

GAURAV AMIN (gga2010)

SAMIR AHMED TALKAL (st4703)

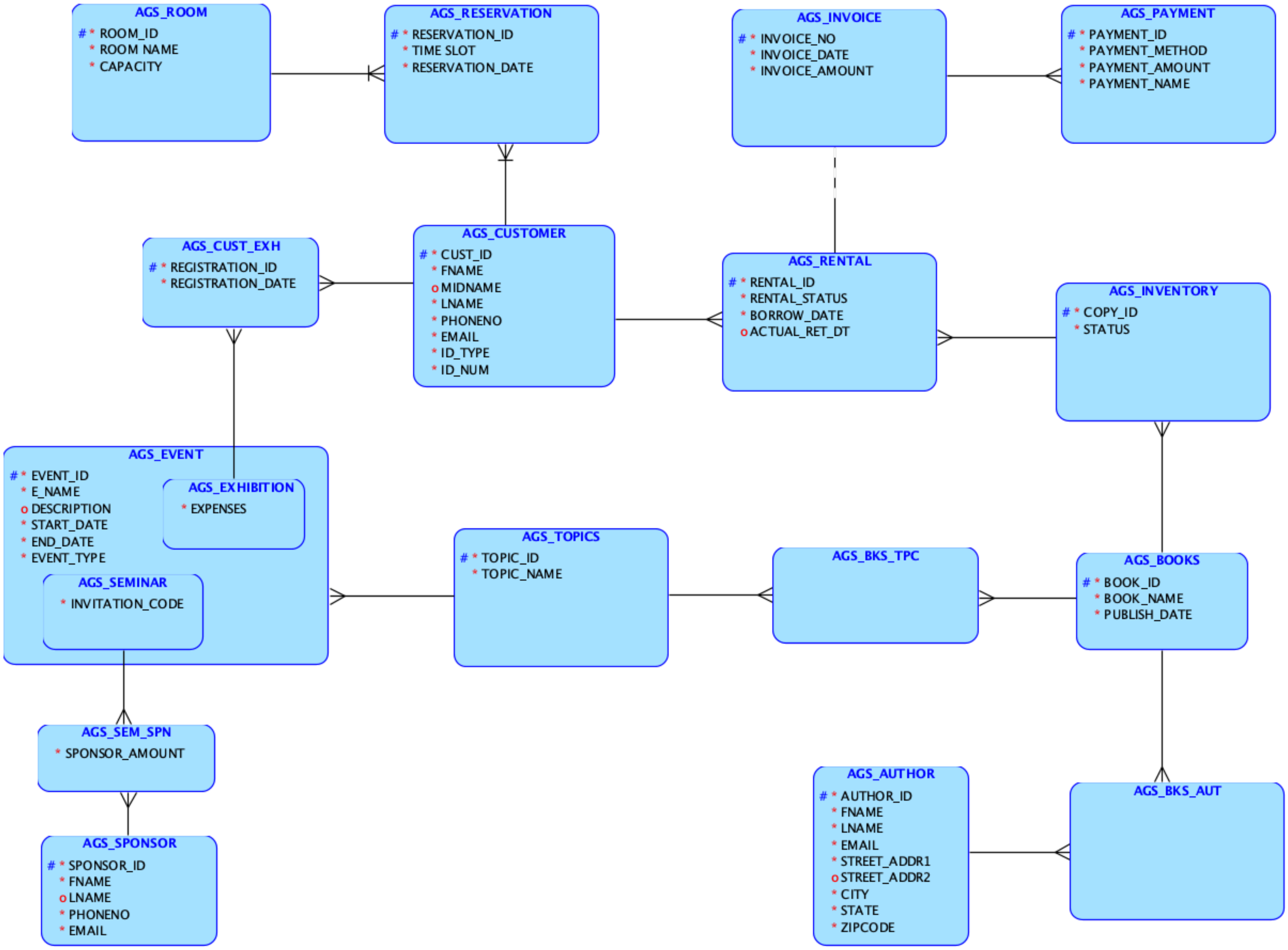
DATE OF SUBMISSION: 11/05/2022

TABLE OF CONTENTS

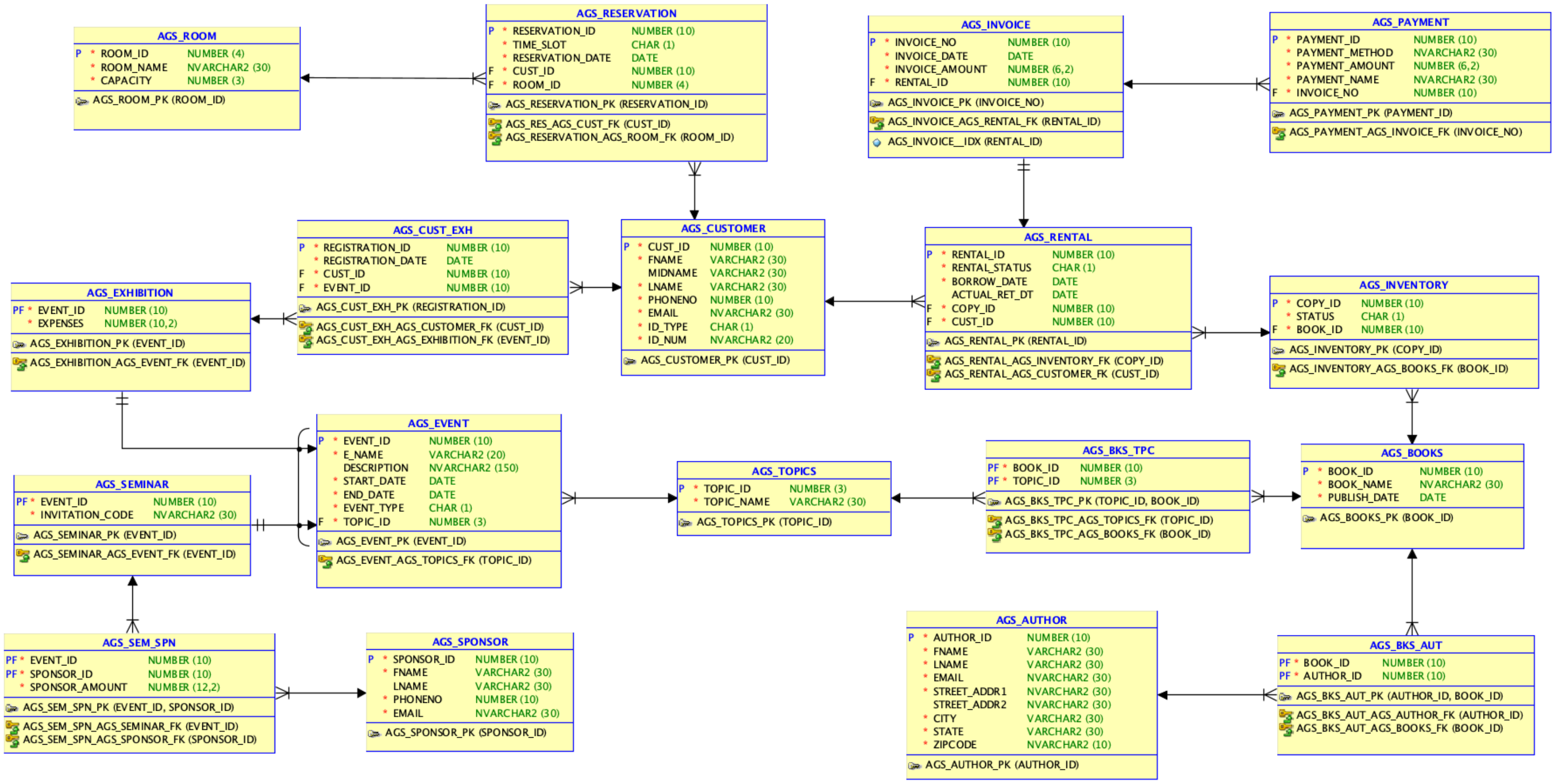
ASSUMPTIONS

- One book can be on many topics and many books can be on one topic.
- One room has many reservations for many time slots.
- One invoice can be for many payments of book.
- Expected return date calculated value.
- Reservation id surrogate key for room ID and customer ID.
- A single copy of a book can be rented by many people at different times.
- Every invoice will have a rental id but every rental service will have an invoice id if the book is returned.
- Every Event will be only on one Topic.
- Every Seminar has a unique invitation ID associated with it
- The author puts in an invitation ID manually and it is checked with the invitation ID given by the seminar so that we come to know if the author is eligible for the seminar. (Hence no intersect between author and Seminar tables as we are not storing attendance)
- Expected Return Date is a calculated field which is calculated as Borrow Date + 1 month.
- Each Seminar will have a unique Invitation Number.
- Reservation Date, Room and Slot should be unique for each reservation.
- Time slots are only one of the 4 vals:
    - ‘1’ for 8am - 10 am
    - ‘2’ for 11am - 1 pm
    - ‘3’ for 2pm - 3pm
    - ‘4’ for 4pm - 6pm

LOGICAL MODEL



RELATIONAL MODEL


AGS_ROOM
P * ROOM_ID NUMBER (4)
* ROOM_NAME NVARCHAR2 (30)
* CAPACITY NUMBER (3)
AGS_ROOM_PK (ROOM_ID)
AGS_RESERVATION
P * RESERVATION_ID NUMBER (10)
* TIME_SLOT CHAR (1)
* RESERVATION_DATE DATE
F * CUST_ID NUMBER (10)
F * ROOM_ID NUMBER (4)
AGS_RESERVATION_PK (RESERVATION_ID)
AGS_RES_AGS_CUST_FK (CUST_ID)
AGS_RESERVATION_AGS_ROOM_FK (ROOM_ID)
AGS_INVOICE
P * INVOICE_NO NUMBER (10)
* INVOICE_DATE DATE
* INVOICE_AMOUNT NUMBER (6,2)
F * RENTAL_ID NUMBER (10)
AGS_INVOICE_PK (INVOICE_NO)
AGS_INVOICE_AGS_RENTAL_FK (RENTAL_ID)
AGS_INVOICE__IDX (RENTAL_ID)
AGS_PAYMENT
P * PAYMENT_ID NUMBER (10)
* PAYMENT_METHOD NVARCHAR2 (30)
* PAYMENT_AMOUNT NUMBER (6,2)
* PAYMENT_NAME NVARCHAR2 (30)
F * INVOICE_NO NUMBER (10)
AGS_PAYMENT_PK (PAYMENT_ID)
AGS_PAYMENT_AGS_INVOICE_FK (INVOICE_NO)
AGS_CUST_EXH
P * REGISTRATION_ID NUMBER (10)
* REGISTRATION_DATE DATE
F * CUST_ID NUMBER (10)
F * EVENT_ID NUMBER (10)
AGS_CUST_EXH_PK (REGISTRATION_ID)
AGS_CUST_EXH_AGS_CUSTOMER_FK (CUST_ID)
AGS_CUST_EXH_AGS_EXHIBITION_FK (EVENT_ID)
AGS_CUSTOMER
P * CUST_ID NUMBER (10)
* FNAME VARCHAR2 (30)
MIDNAME VARCHAR2 (30)
* LNAME VARCHAR2 (30)
* PHONENO NUMBER (10)
* EMAIL NVARCHAR2 (30)
* ID_TYPE CHAR (1)
* ID_NUM NVARCHAR2 (20)
AGS_CUSTOMER_PK (CUST_ID)
AGS_RENTAL
P * RENTAL_ID NUMBER (10)
* RENTAL_STATUS CHAR (1)
* BORROW_DATE DATE
ACTUAL_RET_DT DATE
F * COPY_ID NUMBER (10)
F * CUST_ID NUMBER (10)
AGS_RENTAL_PK (RENTAL_ID)
AGS_RENTAL_AGS_INVENTORY_FK (COPY_ID)
AGS_RENTAL_AGS_CUSTOMER_FK (CUST_ID)
AGS_EXHIBITION
PF * EVENT_ID NUMBER (10)
* EXPENSES NUMBER (10,2)
AGS_EXHIBITION_PK (EVENT_ID)
AGS_EXHIBITION_AGS_EVENT_FK (EVENT_ID)
AGS_INVENTORY
P * COPY_ID NUMBER (10)
* STATUS CHAR (1)
F * BOOK_ID NUMBER (10)
AGS_INVENTORY_PK (COPY_ID)
AGS_INVENTORY_AGS_BOOKS_FK (BOOK_ID)
AGS_EVENT
P * EVENT_ID NUMBER (10)
* E_NAME VARCHAR2 (20)
DESCRIPTION NVARCHAR2 (150)
* START_DATE DATE
* END_DATE DATE
* EVENT_TYPE CHAR (1)
F * TOPIC_ID NUMBER (3)
AGS_EVENT_PK (EVENT_ID)
AGS_EVENT_AGS_TOPICS_FK (TOPIC_ID)
AGS_SEMINAR
PF * EVENT_ID NUMBER (10)
* INVITATION_CODE NVARCHAR2 (30)
AGS_SEMINAR_PK (EVENT_ID)
AGS_SEMINAR_AGS_EVENT_FK (EVENT_ID)
AGS_TOPICS
P * TOPIC_ID NUMBER (3)
* TOPIC_NAME VARCHAR2 (30)
AGS_TOPICS_PK (TOPIC_ID)
AGS_BKS_TPC
PF * BOOK_ID NUMBER (10)
PF * TOPIC_ID NUMBER (3)
AGS_BKS_TPC_PK (TOPIC_ID, BOOK_ID)
AGS_BKS_TPC_AGS_TOPICS_FK (TOPIC_ID)
AGS_BKS_TPC_AGS_BOOKS_FK (BOOK_ID)
AGS_BOOKS
P * BOOK_ID NUMBER (10)
* BOOK_NAME NVARCHAR2 (30)
* PUBLISH_DATE DATE
AGS_BOOKS_PK (BOOK_ID)
AGS_SEM_SPN
PF * EVENT_ID NUMBER (10)
PF * SPONSOR_ID NUMBER (10)
* SPONSOR_AMOUNT NUMBER (12,2)
AGS_SEM_SPN_PK (EVENT_ID, SPONSOR_ID)
AGS_SEM_SPN_AGS_SEMINAR_FK (EVENT_ID)
AGS_SEM_SPN_AGS_SPONSOR_FK (SPONSOR_ID)
AGS_SPONSOR
P * SPONSOR_ID NUMBER (10)
* FNAME VARCHAR2 (30)
LNAME VARCHAR2 (30)
* PHONENO NUMBER (10)
* EMAIL NVARCHAR2 (30)
AGS_SPONSOR_PK (SPONSOR_ID)
AGS_AUTHOR
P * AUTHOR_ID NUMBER (10)
* FNAME VARCHAR2 (30)
* LNAME VARCHAR2 (30)
* EMAIL NVARCHAR2 (30)
* STREET_ADDR1 NVARCHAR2 (30)
STREET_ADDR2 NVARCHAR2 (30)
* CITY VARCHAR2 (30)
* STATE VARCHAR2 (30)
* ZIPCODE NVARCHAR2 (10)
AGS_AUTHOR_PK (AUTHOR_ID)
AGS_BKS_AUT
PF * BOOK_ID NUMBER (10)
PF * AUTHOR_ID NUMBER (10)
AGS_BKS_AUT_PK (AUTHOR_ID, BOOK_ID)
AGS_BKS_AUT_AGS_AUTHOR_FK (AUTHOR_ID)
AGS_BKS_AUT_AGS_BOOKS_FK (BOOK_ID)

DDL SQL CODE

```
-- SQLINES DEMO *** le SQL Developer Data Modeler 22.2.0.165.1149
-- SQLINES DEMO *** -11-03 15:55:15 GMT-04:00
-- SQLINES DEMO *** le Database 21c
-- SQLINES DEMO *** le Database 21c



-- SQLINES DEMO *** no DDL - MDSYS.SDO_GEOMETRY

-- SQLINES DEMO *** no DDL - XMLTYPE

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_author (
   author_id    BIGINT NOT NULL COMMENT 'UNIQUE AUTHOR ID',
   fname        VARCHAR(30) NOT NULL COMMENT 'AUTHOR FIRST NAME',
   lname        VARCHAR(30) NOT NULL COMMENT 'AUTHOR LAST NAME',
   email        NVARCHAR(30) NOT NULL COMMENT 'AUTHOR EMAIL ID',
   street_addr1 NVARCHAR(30) NOT NULL COMMENT 'AUTHOR STREET ADDRESS1',
   street_addr2 NVARCHAR(30) COMMENT 'AUTHOR STREET ADDRESS2',
   city         VARCHAR(30) NOT NULL COMMENT 'AUTHOR CITY ADDRESS',
   state        VARCHAR(30) NOT NULL COMMENT 'AUTHOR STATE',
   zipcode      NVARCHAR(10) NOT NULL COMMENT 'AUTHOR ZIPCODE ADDRESS'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.author_id IS
   'UNIQUE AUTHOR ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.fname IS
   'AUTHOR FIRST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.lname IS
   'AUTHOR LAST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.email IS
   'AUTHOR EMAIL ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.street_addr1 IS
   'AUTHOR STREET ADDRESS1'; */
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.street_addr2 IS
    'AUTHOR STREET ADDRESS2'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.city IS
    'AUTHOR CITY ADDRESS'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.state IS
    'AUTHOR STATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_author.zipcode IS
    'AUTHOR ZIPCODE ADDRESS'; */

ALTER TABLE ags_author ADD CONSTRAINT ags_author_pk PRIMARY KEY ( author_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_bks_aut (
    book_id   BIGINT NOT NULL,
    author_id BIGINT NOT NULL
);

ALTER TABLE ags_bks_aut ADD CONSTRAINT ags_bks_aut_pk PRIMARY KEY (
author_id,
                                                            book_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_bks_tpc (
    book_id  BIGINT NOT NULL,
    topic_id SMALLINT NOT NULL
);

ALTER TABLE ags_bks_tpc ADD CONSTRAINT ags_bks_tpc_pk PRIMARY KEY ( topic_id,
                                                            book_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_books (
    book_id      BIGINT NOT NULL COMMENT 'UNIQUE BOOK ID',
    book_name    NVARCHAR(30) NOT NULL COMMENT 'BOOK NAME',
    publish_date DATETIME NOT NULL COMMENT 'BOOK PUBLISH DATE'
);

/* Moved to CREATE TABLE
```

```
COMMENT ON COLUMN ags_books.book_id IS
    'UNIQUE BOOK ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_books.book_name IS
    'BOOK NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_books.publish_date IS
    'BOOK PUBLISH DATE'; */

ALTER TABLE ags_books ADD CONSTRAINT ags_books_pk PRIMARY KEY ( book_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_cust_exh (
    registration_id   BIGINT NOT NULL COMMENT 'UNIQUE REGISTRATION ID FOR
EACH CUSTOMER FOR EACH EXHIBITION',
    registration_date DATETIME NOT NULL COMMENT 'REGISTRATION DATE',
    cust_id           BIGINT NOT NULL,
    event_id          BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_cust_exh.registration_id IS
    'UNIQUE REGISTRATION ID FOR EACH CUSTOMER FOR EACH EXHIBITION'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_cust_exh.registration_date IS
    'REGISTRATION DATE'; */

ALTER TABLE ags_cust_exh ADD CONSTRAINT ags_cust_exh_pk PRIMARY KEY (
registration_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_customer (
    cust_id BIGINT NOT NULL COMMENT 'UNIQUE CUSTOMER ID',
    fname   VARCHAR(30) NOT NULL COMMENT 'CUSTOMER FIRST NAME',
    midname VARCHAR(30) COMMENT 'CUSTOMER MID NAME',
    lname   VARCHAR(30) NOT NULL COMMENT 'CUSTOMER LAST NAME',
    phoneno BIGINT NOT NULL COMMENT 'CUSTOMER US PHONE NUMBER',
    email   NVARCHAR(30) NOT NULL COMMENT 'CUSTOMER EMAIL ID',
    id_type CHAR(1) NOT NULL COMMENT 'CUSTOMER ID TYPE IN "P"(PASSPORT),
"S"(SSN) OR "D"(DRIVING LICENSE)',
    id_num  NVARCHAR(20) NOT NULL COMMENT 'CUSTOMER ID NUMBER'
);
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.cust_id IS
   'UNIQUE CUSTOMER ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.fname IS
   'CUSTOMER FIRST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.midname IS
   'CUSTOMER MID NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.lname IS
   'CUSTOMER LAST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.phoneno IS
   'CUSTOMER US PHONE NUMBER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.email IS
   'CUSTOMER EMAIL ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.id_type IS
   'CUSTOMER ID TYPE IN "P"(PASSPORT), "S"(SSN) OR "D"(DRIVING LICENSE)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_customer.id_num IS
   'CUSTOMER ID NUMBER'; */

ALTER TABLE ags_customer ADD CONSTRAINT ags_customer_pk PRIMARY KEY (
cust_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_event (
   event_id    BIGINT NOT NULL COMMENT 'UNIQUE EVENT ID',
   e_name      VARCHAR(20) NOT NULL COMMENT 'EVENT NAME',
   description NVARCHAR(150) COMMENT 'EVENT DESCRIPTION',
   start_date  DATETIME NOT NULL COMMENT 'EVENT START DATE',
   end_date    DATETIME NOT NULL COMMENT 'EVENT END DATE',
   event_type  CHAR(1) NOT NULL COMMENT 'EVENT TYPE "S"(SEMINAR) OR
"E"(EXHIBITION)',
   topic_id    SMALLINT NOT NULL
);
```

```
ALTER TABLE ags_event
   ADD CONSTRAINT ch_inh_ags_event CHECK ( event_type IN ( 'E', 'S' ) );

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.event_id IS
   'UNIQUE EVENT ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.e_name IS
   'EVENT NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.description IS
   'EVENT DESCRIPTION'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.start_date IS
   'EVENT START DATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.end_date IS
   'EVENT END DATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_event.event_type IS
   'EVENT TYPE "S"(SEMINAR) OR "E"(EXHIBITION)'; */

ALTER TABLE ags_event ADD CONSTRAINT ags_event_pk PRIMARY KEY ( event_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_exhibition (
   event_id BIGINT NOT NULL COMMENT 'UNIQUE EVENT ID',
   expenses DECIMAL(10, 2) NOT NULL COMMENT 'EXHIBITION EXPENSES IN USD'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_exhibition.event_id IS
   'UNIQUE EVENT ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_exhibition.expenses IS
   'EXHIBITION EXPENSES IN USD'; */

ALTER TABLE ags_exhibition ADD CONSTRAINT ags_exhibition_pk PRIMARY KEY (
event_id );
```

```
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_inventory (
   copy_id BIGINT NOT NULL COMMENT 'UNIQUE COPY ID',
   status  CHAR(1) NOT NULL COMMENT 'COPY STATUS "A"(AVAILABLE) OR
"U"(UNAVAILABLE)',
   book_id BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_inventory.copy_id IS
   'UNIQUE COPY ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_inventory.status IS
   'COPY STATUS "A"(AVAILABLE) OR "U"(UNAVAILABLE)'; */

ALTER TABLE ags_inventory ADD CONSTRAINT ags_inventory_pk PRIMARY KEY (
copy_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_invoice (
   invoice_no   BIGINT NOT NULL COMMENT 'UNIQUE INVOICE NUMBER',
   invoice_date DATETIME NOT NULL COMMENT 'INVOICE GENERATION DATE',
   invoice_amount BIGINT NOT NULL COMMENT 'INVOICE AMOUNT',
   rental_id    BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_invoice.invoice_no IS
   'UNIQUE INVOICE NUMBER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_invoice.invoice_date IS
   'INVOICE GENERATION DATE'; */

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX ags_invoice__idx ON
   ags_invoice (
      rental_id
   ASC );

ALTER TABLE ags_invoice ADD CONSTRAINT ags_invoice_pk PRIMARY KEY (
invoice_no );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
```

```
CREATE TABLE ags_payment (
    payment_id     BIGINT NOT NULL COMMENT 'UNIQUE PAYMENT ID',
    payment_method NVARCHAR(30) NOT NULL COMMENT 'PAYMENT METHOD FOR
RENTAL',
    payment_amount DECIMAL(6, 2) NOT NULL COMMENT 'PAYMENT AMOUNT FOR
RENTAL IN USD',
    payment_name   NVARCHAR(30) NOT NULL COMMENT 'CARD HOLDER NAME OR
NAME OF PERSON WHO MADE PAYMENT',
    invoice_no     BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_payment.payment_id IS
    'UNIQUE PAYMENT ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_payment.payment_method IS
    'PAYMENT METHOD FOR RENTAL'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_payment.payment_amount IS
    'PAYMENT AMOUNT FOR RENTAL IN USD'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_payment.payment_name IS
    'CARD HOLDER NAME OR NAME OF PERSON WHO MADE PAYMENT'; */

ALTER TABLE ags_payment ADD CONSTRAINT ags_payment_pk PRIMARY KEY (
payment_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_rental (
    rental_id     BIGINT NOT NULL COMMENT 'UNIQUE RENTAL ID',
    rental_status CHAR(1) NOT NULL COMMENT 'RENTAL STATUS: "B" BORROWED, "R"
RETURNED OR "L" LATE',
    borrow_date   DATETIME NOT NULL COMMENT 'RENTAL BORROW DATE',
    actual_ret_dt DATETIME COMMENT 'ACTUAL RENTAL RETURN DATE',
    copy_id       BIGINT NOT NULL,
    cust_id       BIGINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_rental.rental_id IS
    'UNIQUE RENTAL ID'; */

/* Moved to CREATE TABLE
```

```
COMMENT ON COLUMN ags_rental.rental_status IS
    'RENTAL STATUS: "B" BORROWED, "R" RETURNED OR "L" LATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_rental.borrow_date IS
    'RENTAL BORROW DATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_rental.actual_ret_dt IS
    'ACTUAL RENTAL RETURN DATE'; */

ALTER TABLE ags_rental ADD CONSTRAINT ags_rental_pk PRIMARY KEY ( rental_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_reservation (
    reservation_id      BIGINT NOT NULL COMMENT 'UNIQUE RESERVATION ID',
    time_slot           CHAR(1) NOT NULL COMMENT 'ONE OF THE 4 TIME SLOTS OF RESERVATION',
    reservation_date    DATETIME NOT NULL COMMENT 'DATE OF RESERVATION',
    cust_id             BIGINT NOT NULL,
    room_id             SMALLINT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_reservation.reservation_id IS
    'UNIQUE RESERVATION ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_reservation.time_slot IS
    'ONE OF THE 4 TIME SLOTS OF RESERVATION'; */

ALTER TABLE ags_reservation ADD CONSTRAINT ags_reservation_pk PRIMARY KEY ( reservation_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_room (
    room_id   SMALLINT NOT NULL COMMENT 'UNIQUE ROOM ID',
    room_name NVARCHAR(30) NOT NULL COMMENT 'NAME OF THE ROOM',
    capacity  SMALLINT NOT NULL COMMENT 'MAXIMUM NUMBER OF PEOPLE IN THE ROOM AT A TIME'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_room.room_id IS
    'UNIQUE ROOM ID'; */
```

```
/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_room.room_name IS
   'NAME OF THE ROOM'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_room.capacity IS
   'MAXIMUM NUMBER OF PEOPLE IN THE ROOM AT A TIME'; */

ALTER TABLE ags_room ADD CONSTRAINT ags_room_pk PRIMARY KEY ( room_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_sem_spn (
   event_id       BIGINT NOT NULL,
   sponsor_id     BIGINT NOT NULL,
   sponsor_amount DECIMAL(12, 2) NOT NULL COMMENT 'SPONSORSHIP AMOUNT IN
USD'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sem_spn.sponsor_amount IS
   'SPONSORSHIP AMOUNT IN USD'; */

ALTER TABLE ags_sem_spn ADD CONSTRAINT ags_sem_spn_pk PRIMARY KEY (
event_id,
                                                          sponsor_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_seminar (
   event_id        BIGINT NOT NULL COMMENT 'UNIQUE EVENT ID',
   invitation_code NVARCHAR(30) NOT NULL COMMENT 'SEMINAR INVITATION
ID/CODE'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_seminar.event_id IS
   'UNIQUE EVENT ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_seminar.invitation_code IS
   'SEMINAR INVITATION ID/CODE'; */

ALTER TABLE ags_seminar ADD CONSTRAINT ags_seminar_pk PRIMARY KEY ( event_id
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_sponsor (
   sponsor_id BIGINT NOT NULL COMMENT 'UNIQUE SPONSOR ID',
   fname      VARCHAR(30) NOT NULL COMMENT 'SPONSOR FIRST NAME',
   lname      VARCHAR(30) COMMENT 'SPONSOR LAST NAME',
   phoneno    BIGINT NOT NULL COMMENT 'SPONSOR PHONE NUMBER',
   email      NVARCHAR(30) NOT NULL COMMENT 'SPONSOR EMAIL'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sponsor.sponsor_id IS
   'UNIQUE SPONSOR ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sponsor.fname IS
   'SPONSOR FIRST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sponsor.lname IS
   'SPONSOR LAST NAME'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sponsor.phoneno IS
   'SPONSOR PHONE NUMBER'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_sponsor.email IS
   'SPONSOR EMAIL'; */

ALTER TABLE ags_sponsor ADD CONSTRAINT ags_sponsor_pk PRIMARY KEY (
sponsor_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ags_topics (
   topic_id   SMALLINT NOT NULL COMMENT 'UNIQUE TOPIC ID',
   topic_name VARCHAR(30) NOT NULL COMMENT 'TOPIC NAME'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_topics.topic_id IS
   'UNIQUE TOPIC ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN ags_topics.topic_name IS
   'TOPIC NAME'; */

ALTER TABLE ags_topics ADD CONSTRAINT ags_topics_pk PRIMARY KEY ( topic_id );
ALTER TABLE ags_bks_aut
   ADD CONSTRAINT ags_bks_aut_ags_author_fk FOREIGN KEY ( author_id )
      REFERENCES ags_author ( author_id );

ALTER TABLE ags_bks_aut
   ADD CONSTRAINT ags_bks_aut_ags_books_fk FOREIGN KEY ( book_id )
      REFERENCES ags_books ( book_id );

ALTER TABLE ags_bks_tpc
   ADD CONSTRAINT ags_bks_tpc_ags_books_fk FOREIGN KEY ( book_id )
      REFERENCES ags_books ( book_id );

ALTER TABLE ags_bks_tpc
   ADD CONSTRAINT ags_bks_tpc_ags_topics_fk FOREIGN KEY ( topic_id )
      REFERENCES ags_topics ( topic_id );

ALTER TABLE ags_cust_exh
   ADD CONSTRAINT ags_cust_exh_ags_customer_fk FOREIGN KEY ( cust_id )
      REFERENCES ags_customer ( cust_id );

ALTER TABLE ags_cust_exh
   ADD CONSTRAINT ags_cust_exh_ags_exhibition_fk FOREIGN KEY ( event_id )
      REFERENCES ags_exhibition ( event_id );

ALTER TABLE ags_event
   ADD CONSTRAINT ags_event_ags_topics_fk FOREIGN KEY ( topic_id )
      REFERENCES ags_topics ( topic_id );

ALTER TABLE ags_exhibition
   ADD CONSTRAINT ags_exhibition_ags_event_fk FOREIGN KEY ( event_id )
      REFERENCES ags_event ( event_id );

ALTER TABLE ags_inventory
   ADD CONSTRAINT ags_inventory_ags_books_fk FOREIGN KEY ( book_id )
      REFERENCES ags_books ( book_id );

ALTER TABLE ags_invoice
   ADD CONSTRAINT ags_invoice_ags_rental_fk FOREIGN KEY ( rental_id )
      REFERENCES ags_rental ( rental_id );

ALTER TABLE ags_payment
   ADD CONSTRAINT ags_payment_ags_invoice_fk FOREIGN KEY ( invoice_no )
      REFERENCES ags_invoice ( invoice_no );

ALTER TABLE ags_rental
   ADD CONSTRAINT ags_rental_ags_customer_fk FOREIGN KEY ( cust_id )
```

```
      REFERENCES ags_customer ( cust_id );

ALTER TABLE ags_rental
   ADD CONSTRAINT ags_rental_ags_inventory_fk FOREIGN KEY ( copy_id )
      REFERENCES ags_inventory ( copy_id );

ALTER TABLE ags_reservation
   ADD CONSTRAINT ags_res_ags_cust_fk FOREIGN KEY ( cust_id )
      REFERENCES ags_customer ( cust_id );

ALTER TABLE ags_reservation
   ADD CONSTRAINT ags_reservation_ags_room_fk FOREIGN KEY ( room_id )
      REFERENCES ags_room ( room_id );

ALTER TABLE ags_sem_spn
   ADD CONSTRAINT ags_sem_spn_ags_seminar_fk FOREIGN KEY ( event_id )
      REFERENCES ags_seminar ( event_id );

ALTER TABLE ags_sem_spn
   ADD CONSTRAINT ags_sem_spn_ags_sponsor_fk FOREIGN KEY ( sponsor_id )
      REFERENCES ags_sponsor ( sponsor_id );

ALTER TABLE ags_seminar
   ADD CONSTRAINT ags_seminar_ags_event_fk FOREIGN KEY ( event_id )
      REFERENCES ags_event ( event_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_2_ags_exhibition_ins;

DELIMITER //

CREATE TRIGGER arc_fkarc_2_ags_exhibition_ins BEFORE INSERT ON ags_exhibition
   FOR EACH ROW

BEGIN
         DECLARE d CHAR(1);
   -- SQLINES LICENSE FOR EVALUATION USE ONLY
   SELECT
      a.event_type
   INTO d
   FROM
      ags_event a
   WHERE
      a.event_id = new.event_id;

   IF ( d IS NULL OR d <> 'E' ) THEN
   SIGNAL SQLSTATE '560BF' SET MESSAGE_TEXT = 'FK
AGS_EXHIBITION_AGS_EVENT_FK in Table AGS_EXHIBITION violates Arc constraint on
Table AGS_EVENT - discriminator column EVENT_TYPE doesn''t have value ''E''';
   END IF;
END;
//
DELIMITER ;

DROP TRIGGER IF EXISTS arc_fkarc_2_ags_exhibition_upd;

DELIMITER //

CREATE TRIGGER arc_fkarc_2_ags_exhibition_upd BEFORE UPDATE ON ags_exhibition
   FOR EACH ROW

BEGIN
         DECLARE d CHAR(1);
   -- SQLINES LICENSE FOR EVALUATION USE ONLY
   SELECT
      a.event_type
   INTO d
   FROM
      ags_event a
   WHERE
      a.event_id = new.event_id;

   IF ( d IS NULL OR d <> 'E' ) THEN
   SIGNAL SQLSTATE '560BF' SET MESSAGE_TEXT = 'FK
AGS_EXHIBITION_AGS_EVENT_FK in Table AGS_EXHIBITION violates Arc constraint on
Table AGS_EVENT - discriminator column EVENT_TYPE doesn''t have value ''E''';
   END IF;
END;
//
DELIMITER ;


-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_2_ags_seminar_ins;

DELIMITER //

CREATE TRIGGER arc_fkarc_2_ags_seminar_ins BEFORE INSERT ON ags_seminar
   FOR EACH ROW

BEGIN
         DECLARE d CHAR(1);
   -- SQLINES LICENSE FOR EVALUATION USE ONLY
   SELECT
      a.event_type
   INTO d
   FROM
      ags_event a
   WHERE
      a.event_id = new.event_id;

   IF ( d IS NULL OR d <> 'S' ) THEN
   SIGNAL SQLSTATE '560BF' SET MESSAGE_TEXT = 'FK
AGS_SEMINAR_AGS_EVENT_FK in Table AGS_SEMINAR violates Arc constraint on Table
AGS_EVENT - discriminator column EVENT_TYPE doesn''t have value ''S''';
   END IF;
END;
//
DELIMITER ;

DROP TRIGGER IF EXISTS arc_fkarc_2_ags_seminar_upd;

DELIMITER //

CREATE TRIGGER arc_fkarc_2_ags_seminar_upd BEFORE UPDATE ON ags_seminar
   FOR EACH ROW

BEGIN
         DECLARE d CHAR(1);
   -- SQLINES LICENSE FOR EVALUATION USE ONLY
   SELECT
      a.event_type
   INTO d
   FROM
      ags_event a
   WHERE
      a.event_id = new.event_id;

   IF ( d IS NULL OR d <> 'S' ) THEN
   SIGNAL SQLSTATE '560BF' SET MESSAGE_TEXT = 'FK
AGS_SEMINAR_AGS_EVENT_FK in Table AGS_SEMINAR violates Arc constraint on Table
AGS_EVENT - discriminator column EVENT_TYPE doesn''t have value ''S''';
   END IF;
END;
//
DELIMITER ;

-- SQLINES DEMO *** per Data Modeler Summary Report:
```

```
--
-- SQLINES DEMO ***            18
-- SQLINES DEMO ***             1
-- SQLINES DEMO ***            37
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO *** DY            0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             2
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***  TYPE         0
-- SQLINES DEMO ***  TYPE         0
-- SQLINES DEMO ***  TYPE BODY        0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO *** EGMENT           0
-- SQLINES DEMO ***             0
-- SQLINES DEMO *** ED VIEW         0
-- SQLINES DEMO *** ED VIEW LOG        0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
--
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
--
-- SQLINES DEMO ***             0
--
-- SQLINES DEMO ***             0
-- SQLINES DEMO *** A           0
-- SQLINES DEMO *** T           0
--
-- SQLINES DEMO ***             0
-- SQLINES DEMO ***             0
```

DDL ADDITIONAL CONSTRAINTS

```
/*MANUALLY ADDED CONSTRAINTS*/
ALTER TABLE AGS_CUSTOMER
ADD CONSTRAINT CHK_cust_id_psd CHECK (id_type = 'P' OR id_type = 'S' OR id_type =
'D');

ALTER TABLE ags_rental
ADD CONSTRAINT CHK_rental_status_BRL CHECK (rental_status = 'B' OR rental_status =
'R' OR rental_status = 'L');

ALTER TABLE ags_inventory ADD CONSTRAINT CHK_STATUS_AU CHECK (status = 'A'
OR status = 'U');

ALTER TABLE ags_reservation
ADD CONSTRAINT CHK_TIME_SLOT_1234 CHECK (time_slot = '1' OR time_slot = '2' OR
time_slot = '3' OR time_slot = '4');

ALTER TABLE ags_reservation
ADD CONSTRAINT CHK_RES_DATE_SLOT_UNQ UNIQUE (reservation_date,time_slot,
room_id);
```

TRIGGER TO GENERATE INVOICE

```
/*TRIGGER TO GENERATE INVOICE*/
DROP TRIGGER IF EXISTS trigger_ags_rental_upd;

DELIMITER //

CREATE TRIGGER trigger_ags_rental_upd BEFORE UPDATE ON ags_rental
   FOR EACH ROW

BEGIN

   IF (new.actual_ret_dt IS NOT NULL) THEN
   BEGIN
   set @actDate = new.actual_ret_dt;
   set @brwDate = (SELECT borrow_date FROM ags_rental where rental_id = new.rental_id);
   set @expDate = (SELECT DATE_ADD(@brwDate, INTERVAL 1 MONTH));
   set @dateDiff = (SELECT DATEDIFF(@expDate,@actDate));
   IF @dateDiff < 0 THEN
   BEGIN
   SET @amt = (SELECT ((DATEDIFF(@expDate,@brwDate)*0.2) +
(DATEDIFF(@actDate,@expDate)*0.4)));
   SET @status = 'L';
   END;
   ELSE
   BEGIN
   SET @amt = (SELECT DATEDIFF(@expDate,@brwDate)*0.2);
   SET @status = 'R';
   END;
   END IF;

   SET @lastInv = (SELECT MAX(INVOICE_NO) FROM ags_invoice);
   INSERT INTO ags_invoice(invoice_no,invoice_date, invoice_amount,rental_id) VALUES
   (IFNULL(@lastInv,0) + 1, NOW(), @amt, new.rental_id);

   SET new.rental_status = @status;

   END;
   END IF;

END;
//
DELIMITER ;
```

DML CODE

```
--Insert initial standing data into Topic table
INSERT INTO ags_topics (topic_id, topic_name) VALUES
('1', 'History'),
('2', 'Children'),
('3', 'Science'),
('4', 'Arts'),
('5', 'Travel'),
('6', 'Adventures'),
('7', 'Drama'),
('8', 'Thriller'),
('9', 'Science Fiction'),
('10', 'Comedy'),
('11', 'Romance'),
('12', 'Crime'),
('13', 'Suspense'),
('14', 'Documentaries'),
('15', 'Food');

/*Customer table insert data*/
INSERT INTO ags_customer (cust_id, fname, midname, lname, phoneno, email, id_type,
id_num) VALUES
('100', 'Rohan','Pratap','Singh',3450000111,'rp11@gmail.com','P','R34232113'),
('101', 'Jack','Barrtet','Jackson',9872224455,'jack_b@gmail.com','D','J432123654567'),
('102', 'Allen','Marrie','Angel',3472225555,'amarrie24@gmail.com','S','123546523'),
('103', 'Barnard','Shay','Lane',9936666221,'bsl1998@gmail.com','D','B654123653875' ),
('104', 'Bell','Ann','Lillian',3845557799,'annbell57@gmail.com','S','432541234'),
('106', 'David','Jeanne', 'Dylan',9328886547,'dd1988@gmail.com','S','543126542'),
('107', 'Disha','Ravi','Singh',3946854321,'dishasingh21@gmail.com','D','D875234454234'),
('108', 'Fuller','Ruth', 'Diego',3776854674,'full16d@gmail.com','D','F567324321657'),
('109', 'Gibson','Faith','Ross',9655567970,'gib20son@gmail.com','P','G45236521'),
('110', 'Hady','Elizabeth','Ruby',3120867464,'elii30@gmail.com','S','565235176'),
('111', 'Kate','Lillie','Colton',9334441124,'kate1986@gmail.com','S','653126895'),
('112', 'Lawrence','Scott','Joseph',9865554422,'lj15@gmail.com','P','L43123531'),
('114', 'Olive','Rae','Nico',3461154002,'olive21@gmail.com','P','O98321412');

INSERT INTO ags_customer (cust_id, fname, lname, phoneno, email, id_type, id_num)
VALUES
('113', 'Mark','Arthur',3485552221,'mark18arthur@gmail.com','D','M234454764342'),
('105', 'Claudia','Isabel',3487775566,'isa18@gmail.com','D','C536324875234');


/*insert data in room table*/
INSERT INTO ags_room(room_id, room_name,capacity) VALUES
('1000','FF0','80'),
('1001','FF1','25'),
('1002','FF2','25'),
('1003','FF3','40'),
('1004','FF4','10'),
('2001','SF1','20'),
('2002','SF2','20'),
('2003','SF3','30'),
('2004','SF4','30'),
('3001','TF1','50'),
('3002','TF2','40'),
('3003','TF3','40'),
('3004','TF4','35'),
('3005','TF5','35'),
('3006','TF6','20'),
('3007','TF7','10');



/*insert data in EVENT table*/
INSERT INTO ags_event (event_id, e_name, description, start_date, end_date, event_type,
topic_id) VALUES
('1', 'Eating Healthy', 'Benefits of eating healthy.', '2022-11-16 11:47:16', '2022-11-17 11:47:16',
'S', '15'),
('2', 'Independence of USA', 'Discussion about the Independence of the USA', '2022-12-11
11:47:16', '2022-12-17 11:47:16', 'S', '1'),
('3', 'Play With Robots', 'Come visit and see top Robots built by top of the line Engineers',
'2022-11-01 11:47:16', '2022-11-01 20:00:16', 'E', '3'),
('4', 'Hikathon', 'Hike various trails within USA and Canada', '2022-12-01 12:00:00', '2022-12-01
01:00:00', 'S', '5'),
('5', 'Art Attack', 'Various Artists put up their work on Display', '2022-11-16', '2022-11-17', 'E',
'4'),
('6', 'Live Love Laugh', 'Open Mic where varrious comics perform.', '2022-12-16 17:00:00',
'2022-11-16 19:00:00', 'S', '10'),
('7', 'Learn Python', 'Learn to use PYTHON', '2022-11-20 11:47:16', '2022-11-20 13:45:00', 'S',
'3'),
('8', 'DBMS Basics', 'Learn the basics of DBMS', '2022-11-21 11:47:16', '2022-11-21 13:45:00',
'S', '3'),
('9', 'Food Fest', 'Come taste food from different parts of the country', '2022-12-20 11:00:00',
'2022-12-21 20:00:00', 'E', '15'),
('10', 'Movie Screening', 'Watch AVATAR 2 coming Saturday', '2022-11-22 18:00:00',
'2022-11-22 21:00:00', 'E', '9'),
('11', 'Advanced Python', 'Learn Advanced Python with us', '2022-11-16 11:00:00', '2022-11-16
12:30:00', 'S', '3'),
('12', 'Coding Bootcamp', 'Learn to code for interview assessments.', '2022-12-01 11:00:00',
'2023-02-25 12:30:00', 'S', '3'),
('13', 'Handicraft Expo', 'Various artists put up their handicrafts for sale.', '2022-11-02 16:00:00',
'2022-11-02 18:00:00', 'E', '4')
;

/*Subtype Data Insertion Start*/
INSERT INTO ags_seminar (event_id, invitation_code) VALUES
('1', 'CODEEH1'),
('2', 'CODEINDIP1'),
('4', 'CODEHIKA1'),
('6', 'CODELLL01'),
('7', 'CODEPYTHON01'),
('8', 'CODEDBMS01'),
('11', 'CODEPYTH02'),
('12', 'CODECOMPCODING01');

INSERT INTO ags_exhibition (event_id, expenses) VALUES
('3', '7000'),
('5', '15000'),
('9', '5000'),
('10', '4500.5'),
('13', '3000.5');
/*Subtype Data Insertion END*/

INSERT INTO ags_cust_exh (registration_id, registration_date, cust_id, event_id) VALUES
('10001', '2022-11-02 12:00:00', '101', '9'),
('10002', '2022-10-01 13:00:00', '107', '13'),
('10003', '2022-11-02 04:00:00', '112', '10'),
('10004', '2022-11-02 15:30:00', '104', '5'),
('10005', '2022-11-02 15:45:00', '109', '9'),
('10006', '2022-11-02 17:23:00', '105', '9'),
('10007', '2022-11-02 09:24:00', '102', '13'),
('10008', '2022-11-02 21:00:00', '100', '10'),
('10009', '2022-11-02 22:34:00', '102', '10'),
('100010', '2022-11-02 18:00:00', '112', '5'),
('100011', '2022-11-02 17:43:00', '114', '13'),
('100012', '2022-11-02 12:23:00', '113', '9'),
('100013', '2022-11-02 14:14:00', '111', '3'),
('100014', '2022-11-02 15:52:00', '110', '9'),
('100015', '2022-11-02 11:12:00', '101', '3'),
('100016', '2022-11-02 19:11:00', '100', '9');


/*Insert into sponsor start*/
INSERT INTO ags_sponsor (sponsor_id, fname, phoneno, email) VALUES
('10000','Techno_Corp',3359553223,'tc@technocorp.inc'),
('10004','AB_Group',9541943813,'ab1@abgroup.inc'),
('10008','Techboom',9925435153,'tb12@techboom.inc'),
('10011','Renault',9321475321,'ab12@renault.inc'),
('10012','Mongo',9231513453,'dbs@mongo.inc'),
('10013','BOFA',9152346431,'ad9@bofa.inc');

INSERT INTO ags_sponsor (sponsor_id, fname,lname, phoneno, email) VALUES
('10001','Matthew','Wade',3856553224,'mwade@gmail.com'),
('10002','Lina','Marie',3439663217,'linam28@gmail.com'),
('10003','Adi','Chan',3985955324,'chanadi2@gmail.com'),
('10005','Gaurang','Sen',3652413535,'gaursen12@gamil.com'),
('10006','Yash','Chauhan',3234562371,'yashc1995@gmail.com'),
('10007','Mark', 'Colton',3098541431,'coltonmark23@gmail.com'),
('10009','Sam','Billings',3654113571,'samb32@gmail.com'),
('10010','Olivia','Shaw',3863511544,'shawoli11@gmail.com'),
('10014','Lauren','Ruth',3542246114,'ruthlauren32@gmail.com');
/*Insert into sponsor end*/

/*Insert into sponsor-event intersect table*/
INSERT INTO ags_sem_spn (event_id, sponsor_id, sponsor_amount) VALUES
('2', '10002', '1000'),
('8', '10012', '1000'),
('7', '10004', '1000'),
('1', '10008', '1000'),
('2', '10001', '1000'),
('11', '10013', '1000'),
('4', '10012', '1000'),
('2', '10014', '1000'),
('12', '10001', '1000'),
('6', '10002', '1000'),
('8', '10010', '1000'),
('2', '10011', '1000'),
('1', '10002', '1000');


-- Insert initial standing data into Author table
INSERT INTO ags_author
        (author_id,fname,lname,email,street_addr1,street_addr2,city,state,zipcode)
VALUES
        (1,'STEPHEN','KING','stephenking@gmail.com','71 ST. NICHOLAS DRIVE','','NORTH
POLE','ALASKA','99705'),
   (2,'STAN','LEE','stanlee@yahoo.com','2200 HWY 98 STE
5','','DAPHNE','ALABAMA','36526'),
   (3,'J.K.','ROWLING','jkrowling@gmail','2528 W 28TH ST','','PINE
BLUFF','ARKANSAS','71603'),
   (4,'JOHN','GRISHAM','johngrisham@yahoo.com','9565 E 22ND ST','SUITE
115','TUCSON','ARIZONA','85748'),
   (5,'R.L.','STINE','rlstine@gmail.com','16232 FOOTHILL
BLVD',",'UPLAND','CALIFORNIA','91768'),
   (6,'JAMES','PATTERSON','jamespatterson@yahoo.com','850 HARTFORD
TNPK','CRYSTAL MALL','WATERFORD','CINCINNATI','06385'),
   (7,'ANNE','RICE','annerice@gmail.com','1630 CONNECTICUT
AVE',",'WASHINGTON','DC','20009'),
   (8,'DEAN','KOONTZ','deankoontz@yahoo.com','3601 CONCORD
PIKE',",'WILMNINGTON','DELAWARE','19702'),
   (9,'DANIELLE','STEEL','daniellesteel@gmail.com','1110 AIRPORT
BOULEVARD',",'PENSACOLA','FLORIDA','32504'),
   (10,'MARY','CLARK','maryclark@yahoo.com','4375 LEXINGTON RD','SUITE
C-4','ATHENS','GEORGIA','30605'),
   (11,'GEORGE','MARTIN','georgemartin@gmail.com','95-1249 MEHEULA PKWY','STE
D-13','MILILANI','HAWAII','96789'),
   (12,'NORA','ROBERTS','noraroberts@yahoo.com','4100 UNIVERSITY AVE','SUITE
114','WEST DES MOINES','ITHACA','50266'),
   (13,'TONI','MORRISON','tonimorrison@gmail.com','2000 W PULLMAN
RD',",'MOSCOW','IDAHO','83843'),
   (14,'CLIVE','BARKER','clivebarker@yahoo.com','729 W MAIN ST',",'WEST
DUNDEE','ILLINOIS','60118'),
   (15,'MARGARET','ATWOOD','margaretatwood@gmail.com','160 MONTAGUE
STREET',",'BROOKLYN','NEW YORK','11201'),
   (16,'DOUGLAS','PRESTON','douglaspreston@yahoo.com','7911 MALL
RD',",'FLORENCE','KENTUCKY','41042'),
          (17,'LINCOLN','CHILD','lincolnchild@gmail.com','4801 OUTER
LOOP',",'LOUISVILLE','KENTUCKY','40219'),
   (18,'CHRIS','GRABENSTEIN','chrisgrabenstein@yahoo.com','61103 AIRPORT
RD',",'SLIDELL','LOS ANGELES','70460');

-- Insert initial standing data into Books table
INSERT INTO ags_books
(book_id,book_name,publish_date)
VALUES
(1,'THE STAND','1978-10-03'),
(2,'THE DEAD ZONE','1979-08-30'),
(3,'BALANCE OF POWER','2017-03-07'),
(4,'THE ICKABOG','2020-11-10'),
(5,'TROUBLED BLOOD','2020-09-15'),
(6,'THE ULTIMATE SPIDER MAN','2011-06-06'),
(7,'A TIME TO KILL','1989-05-09'),
(8,'THE FIRM','1991-02-01'),
(9,'THE WRONG NUMBER','1990-09-08'),
(10,'I FUNNY','2012-08-03'),
(11,'THE HOUSE OF KENNEDY','2020-04-13'),
(12,'A GAME OF THRONES','1996-02-01'),
(13,'THE TESTAMENTS','2019-09-10'),
(14,'LEGACY','2021-05-25'),
(15,'VISION IN WHITE','2009-04-28'),
(16,'RELIC','1996-01-15');

-- Insert initial standing data into books and author intercept table
INSERT INTO ags_bks_aut
(book_id,author_id)
VALUES
(4,2),
(5,2),
(14,14),
(9,5),
(11,6),
(10,6),
(10,18),
(4,3),
(16,16),
(16,17);

-- Insert initial standing data into books and topic intercept table
INSERT INTO ags_bks_tpc
(book_id,topic_id)
VALUES
(3,8),
(3,13),
(5,8),
(5,12),
(6,9),
(9,2),
(9,8),
(9,13),
(15,11),
(16,8),
(16,9);


INSERT INTO ags_inventory (copy_id, status, book_id) VALUES
('101', 'A', '1'),
('102', 'A', '1'),
('103', 'U', '1'),
('104', 'A', '12'),
('105', 'A', '11'),
('106', 'U', '13'),
('107', 'A', '14'),
('108', 'A', '16'),
('109', 'U', '3'),
('110', 'A', '4'),
('111', 'U', '5'),
('112', 'A', '1'),
('113', 'A', '8'),
('114', 'U', '9'),
('115', 'A', '11'),
('116', 'A', '15'),
('117', 'U', '10'),
('118', 'A', '12');


/*RENTAL WILL ONLY INSERT VALUES WITH NO ACTUAL_RET DATE. ACTUAL RET
DATE WILL BE ONLY POPUILATED IN UPDATES*/
INSERT INTO ags_rental (rental_id, rental_status, borrow_date, actual_ret_dt, copy_id, cust_id)
VALUES
('101', 'B', '2022-11-01 13:12:00', NULL, '115', '102'),
('102', 'B', '2022-12-01 13:12:00', NULL, '108', '108'),
('103', 'B', '2022-01-23 15:12:00', NULL, '104', '111'),
('104', 'B', '2022-02-01 13:12:00', NULL, '101', '101'),
('105', 'B', '2022-03-21 23:12:00', NULL, '114', '102'),
('106', 'B', '2022-04-01 13:12:00', NULL, '111', '111'),
('107', 'B', '2022-05-11 13:12:00', NULL, '112', '102'),
('108', 'B', '2022-11-01 13:12:00', NULL, '101', '103'),
('109', 'B', '2022-11-01 11:12:00', NULL, '115', '106'),
('110', 'B', '2022-10-23 13:12:00', NULL, '103', '109'),
('111', 'B', '2022-11-01 10:12:00', NULL, '109', '111'),
('112', 'B', '2022-03-13 13:12:00', NULL, '114', '107'),
('113', 'B', '2022-10-01 09:12:00', NULL, '109', '113'),
('114', 'B', '2022-08-01 13:12:00', NULL, '109', '107'),
('115', 'B', '2022-11-11 13:12:00', NULL, '114', '111'),
('116', 'B', '2022-09-01 15:12:00', NULL, '104', '101'),
('117', 'B', '2022-11-17 13:12:00', NULL, '109', '107'),
('118', 'B', '2022-01-01 20:12:00', NULL, '114', '106'),
('119', 'B', '2022-04-18 21:12:00', NULL, '111', '110'),
('120', 'B', '2022-05-13 22:12:00', NULL, '112', '104');


/*UPDATES IN RENTAL TABLE ACTUAL_RET_DATE WILL TRIGGER INSERT VALUES
IN INVOICE TABLE
SO, INSTEAD OF INSERTING IN INVOICE, WRITING MODIFY QUERY ON RENTAL
TABLE*/
UPDATE ags_rental
SET actual_ret_dt = '2022-12-03 13:12:00'
WHERE rental_id in (101,102,103,104,105,106,107,108,109,110,111,112)
```

```
INSERT INTO ags_payment (payment_id, payment_method, payment_amount, payment_name,
invoice_no) VALUES
 ('1001', 'Cash', '6.80', 'Adi', '1'),
 ('1002', 'Credit Card', '6.20', 'Allen''s credit card', '2'),
 ('1003', 'Paypal', '59.4', 'Kate''s Paypal', '3'),
 ('1004', 'Cash', '30', 'Kate', '3'),
 ('1005', 'Credit Card', '30', 'Kate''s Credit Card', '3'),
 ('1006', 'Cash', '116.4', 'Jack', '4'),
 ('1007', 'Credit Card', '96.60', 'Allen''s credit card', '5'),
 ('1009', 'Paypal', '36.20', 'Barnard', '7'),
 ('1010', 'Credit Card', '30', 'Barnard ''s Credit Card', '7'),
 ('1011', 'Cash', '10', 'Barnard', '7');

 INSERT INTO ags_reservation (reservation_id, time_slot, cust_id, room_id, reservation_date)
VALUES
 ('1001', '1', '101', '2003','2022-11-05'),
 ('1002', '2', '103', '1004','2022-10-12'),
 ('1003', '3', '113', '1000','2022-10-13'),
 ('1004', '4', '104', '2003','2022-11-14'),
 ('1005', '2', '105', '3001','2022-09-16'),
 ('1006', '4', '111', '3002','2022-11-23'),
 ('1007', '2', '111', '3003','2022-07-25'),
 ('1008', '1', '114', '3005','2022-11-28'),
 ('1009', '1', '104', '2003','2022-12-05'),
 ('1010', '4', '103', '2001','2022-11-05'),
 ('1011', '4', '108', '2002','2022-11-05'),
 ('1012', '1', '107', '2003','2022-01-05'),
 ('1013', '2', '109', '3002','2022-11-06'),
 ('1014', '3', '110', '3005','2022-11-02');x
```

SELECT QUERY OUTPUTS

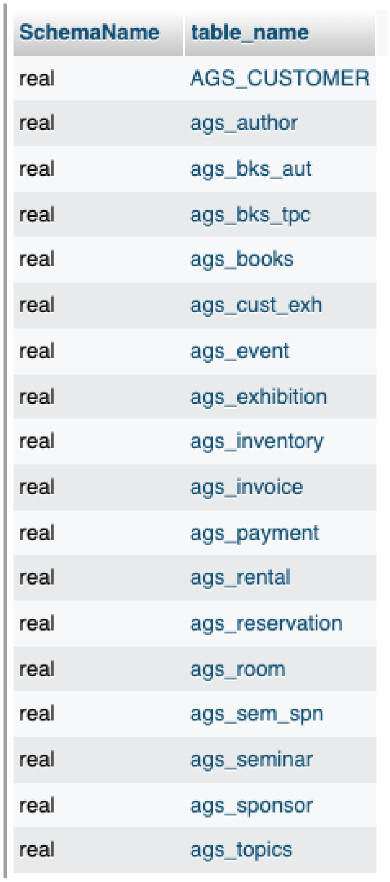
```
select count(1) from ags_topics;
15

select count(1) from ags_customer;
16

select count(1) from ags_room;
16

select count(1) from ags_event;
13

/*Since seminar and exhibition are subtypes, they have less initial/standing data than.*/
select count(1) from ags_seminar;
8

select count(1) from ags_exhibition;
5

select count(1) from ags_cust_exh;
16

select count(1) from ags_sponsor;
15

select count(1) from ags_author;
18

select count(1) from ags_books;
16

select count(1) from ags_bks_aut;
10

select count(1) from ags_rental;
20

select count(1) from ags_bks_tpc;
11

select count(1) from ags_sem_spn;
13

SELECT count(1) from ags_inventory;
```

```
18

select count(1) from ags_invoice;
12


select count(1) from ags_payment;
10

select count(1) from ags_reservation;
14
```

DATA DICTIONARY QUERIES AND OUTPUTS

1.

```
SELECT table_schema AS SchemaName, table_name
  FROM INFORMATION_SCHEMA.TABLES
```



SchemaName	table_name
real	AGS_CUSTOMER
real	ags_author
real	ags_bks_aut
real	ags_bks_tpc
real	ags_books
real	ags_cust_exh
real	ags_event
real	ags_exhibition
real	ags_inventory
real	ags_invoice
real	ags_payment
real	ags_rental
real	ags_reservation
real	ags_room
real	ags_sem_spn
real	ags_seminar
real	ags_sponsor
real	ags_topics

2.

```
select t.table_name,t.schema_name,t.column_name,
GROUP_CONCAT(t.constraint_type ORDER BY t.constraint_type SEPARATOR ',') AS
CONSTRAINTS,
t.optional,t.column_type,t.data_type,t.column_comment
from
(
select 'REAL' as schema_name, c.TABLE_NAME, c.COLUMN_NAME, c.IS_NULLABLE AS
OPTIONAL, c.DATA_TYPE, c.COLUMN_TYPE, cons.constraint_type,
c.COLUMN_COMMENT
from information_schema.columns c
LEFT JOIN
(
SELECT
tc.table_name,tc.constraint_schema,tc.table_schema,kcu.column_name,tc.constraint_type FROM
information_schema.table_constraints tc
INNER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE kcu
on kcu.table_schema = tc.table_schema
AND kcu.table_name = tc.table_name
and kcu.constraint_name = tc.constraint_name
where tc.table_schema = 'REAL'
GROUP BY tc.table_schema,tc.table_name, kcu.column_name,tc.constraint_type
) cons
on c.table_schema = cons.table_schema
AND c.table_name = cons.table_name
and c.column_name = cons.column_name
where c.table_schema = 'REAL'
) t
group by t.schema_name,t.table_name,t.column_name;
```

TABLE_NAME	schema_name	COLUMN_NAME	CONSTRAINTS	OPTIONAL	COLUMN_TYPE	COLUMN_COMMENT
ags_author	REAL	author_id	PRIMARY KEY	NO	bigint(20)	UNIQUE AUTHOR ID
ags_author	REAL	city		NO	varchar(30)	AUTHOR CITY ADDRESS
ags_author	REAL	email		NO	varchar(30)	AUTHOR EMAIL ID
ags_author	REAL	fname		NO	varchar(30)	AUTHOR FIRST NAME
ags_author	REAL	lname		NO	varchar(30)	AUTHOR LAST NAME
ags_author	REAL	state		NO	varchar(30)	AUTHOR STATE
ags_author	REAL	street_addr1		NO	varchar(30)	AUTHOR STREET ADDRESS1
ags_author	REAL	street_addr2		YES	varchar(30)	AUTHOR STREET ADDRESS2
ags_author	REAL	zipcode		NO	varchar(10)	AUTHOR ZIPCODE ADDRESS
ags_bks_aut	REAL	author_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	
ags_bks_aut	REAL	book_id	FOREIGN KEY,PRI	NO	bigint(20)	

			MARY KEY			
ags_bks_tpc	REAL	book_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	
ags_bks_tpc	REAL	topic_id	FOREIGN KEY,PRIMARY KEY	NO	smallint(6)	
ags_books	REAL	book_id	PRIMARY KEY	NO	bigint(20)	UNIQUE BOOK ID
ags_books	REAL	book_name		NO	varchar(30)	BOOK NAME
ags_books	REAL	publish_date		NO	datetime	BOOK PUBLISH DATE
AGS_CUSTOMER	REAL	cust_id	PRIMARY KEY	NO	bigint(20)	UNIQUE CUSTOMER ID
AGS_CUSTOMER	REAL	email		NO	varchar(30)	CUSTOMER EMAIL ID
AGS_CUSTOMER	REAL	fname		NO	varchar(30)	CUSTOMER FIRST NAME
AGS_CUSTOMER	REAL	id_num		NO	varchar(20)	CUSTOMER ID NUMBER
AGS_CUSTOMER	REAL	id_type		NO	char(1)	CUSTOMER ID TYPE IN 'P'(PASSPORT), 'S'(SSN) OR 'D'(DRIVING LICENSE)
AGS_CUSTOMER	REAL	lname		NO	varchar(30)	CUSTOMER LAST NAME
AGS_CUSTOMER	REAL	midname		YES	varchar(30)	CUSTOMER MID NAME
AGS_CUSTOMER	REAL	phoneno		NO	bigint(20)	CUSTOMER US PHONE NUMBER
ags_cust_exh	REAL	cust_id	FOREIGN KEY	NO	bigint(20)	
ags_cust_exh	REAL	event_id	FOREIGN KEY	NO	bigint(20)	

ags_cust_exh	REAL	registration_date		NO	datetime	REGISTRATION DATE
ags_cust_exh	REAL	registration_id	PRIMARY KEY	NO	bigint(20)	UNIQUE REGISTRATION ID FOR EACH CUSTOMER FOR EACH EXHIBITION
ags_event	REAL	description		YES	varchar(150)	EVENT DESCRIPTION
ags_event	REAL	end_date		NO	datetime	EVENT END DATE
ags_event	REAL	event_id	PRIMARY KEY	NO	bigint(20)	UNIQUE EVENT ID
ags_event	REAL	event_type		NO	char(1)	EVENT TYPE 'S'(SEMINAR) OR 'E'(EXHIBITION)
ags_event	REAL	e_name		NO	varchar(20)	EVENT NAME
ags_event	REAL	start_date		NO	datetime	EVENT START DATE
ags_event	REAL	topic_id	FOREIGN KEY	NO	smallint(6)	
ags_exhibition	REAL	event_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	UNIQUE EVENT ID
ags_exhibition	REAL	expenses		NO	decimal(10,2)	EXHIBITION EXPENSES IN USD
ags_inventory	REAL	book_id	FOREIGN KEY	NO	bigint(20)	
ags_inventory	REAL	copy_id	PRIMARY KEY	NO	bigint(20)	UNIQUE COPY ID
ags_inventory	REAL	status		NO	char(1)	COPY STATUS 'A'(AVAILABLE) OR 'U'(UNAVAILABLE)
ags_invoice	REAL	invoice_amount		NO	decimal(6,2)	INVOICE AMOUNT
ags_invoice	REAL	invoice_date		NO	datetime	INVOICE GENERATION DATE
ags_invoice	REAL	invoice_no	PRIMARY KEY	NO	bigint(20)	UNIQUE INVOICE NUMBER

ags_invoice	REAL	rental_id	FOREIGN KEY,UNIQUE	NO	bigint(20)	
ags_payment	REAL	invoice_no	FOREIGN KEY	NO	bigint(20)	
ags_payment	REAL	payment_amount		NO	decimal(6,2)	PAYMENT AMOUNT FOR RENTAL IN USD
ags_payment	REAL	payment_id	PRIMARY KEY	NO	bigint(20)	UNIQUE PAYMENT ID
ags_payment	REAL	payment_method		NO	varchar(30)	PAYMENT METHOD FOR RENTAL
ags_payment	REAL	payment_name		NO	varchar(30)	CARD HOLDER NAME OR NAME OF PERSON WHO MADE PAYMENT
ags_rental	REAL	actual_ret_dt		YES	datetime	ACTUAL RENTAL RETURN DATE
ags_rental	REAL	borrow_date		NO	datetime	RENTAL BORROW DATE
ags_rental	REAL	copy_id	FOREIGN KEY	NO	bigint(20)	
ags_rental	REAL	cust_id	FOREIGN KEY	NO	bigint(20)	
ags_rental	REAL	rental_id	PRIMARY KEY	NO	bigint(20)	UNIQUE RENTAL ID
ags_rental	REAL	rental_status		NO	char(1)	RENTAL STATUS: 'B' BORROWED, 'R' RETURNED OR 'L' LATE
ags_reservation	REAL	cust_id	FOREIGN KEY	NO	bigint(20)	
ags_reservation	REAL	reservation_id	PRIMARY KEY	NO	bigint(20)	UNIQUE RESERVATION ID
ags_reservation	REAL	reservation_date		NO	datetime	DATE OF RESERVATION
ags_reservation	REAL	room_id	FOREIGN KEY	NO	smallint(6)	

ags_reservation	REAL	time_slot		NO	char(1)	ONE OF THE 4 TIME SLOTS OF RESERVATION
ags_room	REAL	capacity		NO	smallint(6)	MAXIMUM NUMBER OF PEOPLE IN THE ROOM AT A TIME
ags_room	REAL	room_id	PRIMARY KEY	NO	smallint(6)	UNIQUE ROOM ID
ags_room	REAL	room_name		NO	varchar(30)	NAME OF THE ROOM
ags_seminar	REAL	event_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	UNIQUE EVENT ID
ags_seminar	REAL	invitation_code		NO	varchar(30)	SEMINAR INVITATION ID/CODE
ags_sem_spn	REAL	event_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	
ags_sem_spn	REAL	sponsor_amount		NO	decimal(12,2)	SPONSORSHIP AMOUNT IN USD
ags_sem_spn	REAL	sponsor_id	FOREIGN KEY,PRIMARY KEY	NO	bigint(20)	
ags_sponsor	REAL	email		NO	varchar(30)	SPONSOR EMAIL
ags_sponsor	REAL	fname		NO	varchar(30)	SPONSOR FIRST NAME
ags_sponsor	REAL	lname		YES	varchar(30)	SPONSOR LAST NAME
ags_sponsor	REAL	phoneno		NO	bigint(20)	SPONSOR PHONE NUMBER
ags_sponsor	REAL	sponsor_id	PRIMARY KEY	NO	bigint(20)	UNIQUE SPONSOR ID
ags_topics	REAL	topic_id	PRIMARY KEY	NO	smallint(6)	UNIQUE TOPIC ID
ags_topics	REAL	topic_name		NO	varchar(30)	TOPIC NAME