-- Generated by Oracle SQL Developer Data Modeler 19.4.0.350.1424

-- at: 2020-06-04 20:40:40 EEST

-- site: Oracle Database 12c

-- type: Oracle Database 12c

CREATE TABLE availiability (

availiability\_code VARCHAR2(30) NOT NULL,

"date" DATE NOT NULL,

departure\_time DATE NOT NULL,

flight\_time VARCHAR2(30) NOT NULL,

number\_business\_seat INTEGER,

number\_economy\_seat INTEGER,

booked\_business INTEGER,

booked\_economy INTEGER,

availiability\_id NUMBER NOT NULL,

reservation\_reservation\_id NUMBER NOT NULL,

flight\_flight\_id NUMBER NOT NULL

);

CREATE UNIQUE INDEX availiability\_\_idx ON

availiability (

reservation\_reservation\_id

ASC );

CREATE UNIQUE INDEX availiability\_\_idxv1 ON

availiability (

flight\_flight\_id

ASC );

ALTER TABLE availiability ADD CONSTRAINT availiability\_pk PRIMARY KEY ( availiability\_id );

CREATE TABLE b\_office (

b\_office\_id NUMBER NOT NULL,

city\_city\_id NUMBER NOT NULL

);

CREATE UNIQUE INDEX b\_office\_\_idx ON

b\_office (

city\_city\_id

ASC );

ALTER TABLE b\_office ADD CONSTRAINT b\_office\_pk PRIMARY KEY ( b\_office\_id );

CREATE TABLE city (

city\_id NUMBER NOT NULL,

b\_office\_b\_office\_id NUMBER NOT NULL,

type\_of\_money VARCHAR2(30) NOT NULL

);

CREATE UNIQUE INDEX city\_\_idx ON

city (

b\_office\_b\_office\_id

ASC );

ALTER TABLE city ADD CONSTRAINT city\_pk PRIMARY KEY ( city\_id );

CREATE TABLE customer (

name VARCHAR2(30) NOT NULL,

surname VARCHAR2(30) NOT NULL,

customer\_id NUMBER NOT NULL,

b\_office\_b\_office\_id NUMBER,

reservation\_reservation\_id NUMBER NOT NULL

);

CREATE UNIQUE INDEX customer\_\_idx ON

customer (

reservation\_reservation\_id

ASC );

ALTER TABLE customer ADD CONSTRAINT customer\_pk PRIMARY KEY ( customer\_id );

CREATE TABLE customer\_reservation (

customer\_customer\_id NUMBER NOT NULL,

reservation\_reservation\_id NUMBER NOT NULL

);

CREATE TABLE email (

customer\_customer\_id NUMBER

);

CREATE TABLE flight (

flight\_code VARCHAR2(30) NOT NULL,

business\_seat VARCHAR2(30) NOT NULL,

airline\_code VARCHAR2(30) NOT NULL,

smokers\_seat INTEGER NOT NULL,

flight\_id NUMBER NOT NULL,

customer\_customer\_id NUMBER NOT NULL,

reservation\_reservation\_id NUMBER NOT NULL,

availiability\_availiability\_id NUMBER NOT NULL

);

CREATE UNIQUE INDEX flight\_\_idx ON

flight (

reservation\_reservation\_id

ASC );

CREATE UNIQUE INDEX flight\_\_idxv1 ON

flight (

availiability\_availiability\_id

ASC );

ALTER TABLE flight ADD CONSTRAINT flight\_pk PRIMARY KEY ( flight\_id );

CREATE TABLE phone\_number (

country\_code INTEGER NOT NULL,

region\_code INTEGER NOT NULL,

local\_number INTEGER NOT NULL,

customer\_customer\_id NUMBER

);

CREATE TABLE postal\_address (

street VARCHAR2(30) NOT NULL,

city VARCHAR2(30) NOT NULL,

country VARCHAR2(30) NOT NULL,

postcodes INTEGER NOT NULL,

customer\_customer\_id NUMBER

);

CREATE TABLE reservation (

reservation\_code VARCHAR2(30) NOT NULL,

reservation\_city VARCHAR2(30),

reservation\_date DATE NOT NULL,

reservation\_flight\_code INTEGER NOT NULL,

departure\_date DATE NOT NULL,

departure\_time DATE NOT NULL,

arrival\_time DATE NOT NULL,

attribute\_8 unknown

-- ERROR: Datatype UNKNOWN is not allowed

NOT NULL,

seat\_category VARCHAR2(30) NOT NULL,

first\_deposit\_money VARCHAR2(30),

other\_money VARCHAR2(30),

seat\_status VARCHAR2(30),

reservation\_id NUMBER NOT NULL,

availiability\_availiability\_id NUMBER NOT NULL,

flight\_flight\_id NUMBER NOT NULL,

ticket\_cost\_ticket\_cost\_id NUMBER NOT NULL,

customer\_customer\_id NUMBER NOT NULL,

b\_office\_b\_office\_id NUMBER

);

CREATE UNIQUE INDEX reservation\_\_idx ON

reservation (

availiability\_availiability\_id

ASC );

CREATE UNIQUE INDEX reservation\_\_idxv1 ON

reservation (

flight\_flight\_id

ASC );

CREATE UNIQUE INDEX reservation\_\_idxv2 ON

reservation (

ticket\_cost\_ticket\_cost\_id

ASC );

CREATE UNIQUE INDEX reservation\_\_idxv3 ON

reservation (

customer\_customer\_id

ASC );

ALTER TABLE reservation ADD CONSTRAINT reservation\_pk PRIMARY KEY ( reservation\_id );

CREATE TABLE ticket\_cost (

ticket\_cost FLOAT(30) NOT NULL,

flight\_cost FLOAT(30) NOT NULL,

business\_cost VARCHAR2(30) NOT NULL,

ticket\_cost\_id NUMBER NOT NULL,

reservation\_reservation\_id NUMBER NOT NULL,

departure\_city\_money VARCHAR2(30) NOT NULL,

arrival\_city\_money VARCHAR2

-- ERROR: VARCHAR2 size not specified

NOT NULL

);

CREATE UNIQUE INDEX ticket\_cost\_\_idx ON

ticket\_cost (

reservation\_reservation\_id

ASC );

ALTER TABLE ticket\_cost ADD CONSTRAINT ticket\_cost\_pk PRIMARY KEY ( ticket\_cost\_id );

ALTER TABLE availiability

ADD CONSTRAINT availiability\_flight\_fk FOREIGN KEY ( flight\_flight\_id )

REFERENCES flight ( flight\_id );

ALTER TABLE availiability

ADD CONSTRAINT availiability\_reservation\_fk FOREIGN KEY ( reservation\_reservation\_id )

REFERENCES reservation ( reservation\_id );

ALTER TABLE b\_office

ADD CONSTRAINT b\_office\_city\_fk FOREIGN KEY ( city\_city\_id )

REFERENCES city ( city\_id );

ALTER TABLE city

ADD CONSTRAINT city\_b\_office\_fk FOREIGN KEY ( b\_office\_b\_office\_id )

REFERENCES b\_office ( b\_office\_id );

ALTER TABLE customer

ADD CONSTRAINT customer\_b\_office\_fk FOREIGN KEY ( b\_office\_b\_office\_id )

REFERENCES b\_office ( b\_office\_id );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE customer\_reservation

ADD CONSTRAINT customer\_reservation\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE customer

ADD CONSTRAINT customer\_reservation\_fk FOREIGN KEY ( reservation\_reservation\_id )

REFERENCES reservation ( reservation\_id );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE customer\_reservation

ADD CONSTRAINT customer\_reservation\_reservation\_fk FOREIGN KEY ( reservation\_reservation\_id )

REFERENCES reservation ( reservation\_id );

ALTER TABLE email

ADD CONSTRAINT email\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE flight

ADD CONSTRAINT flight\_availiability\_fk FOREIGN KEY ( availiability\_availiability\_id )

REFERENCES availiability ( availiability\_id );

ALTER TABLE flight

ADD CONSTRAINT flight\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE flight

ADD CONSTRAINT flight\_reservation\_fk FOREIGN KEY ( reservation\_reservation\_id )

REFERENCES reservation ( reservation\_id );

ALTER TABLE phone\_number

ADD CONSTRAINT phone\_number\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE postal\_address

ADD CONSTRAINT postal\_address\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE reservation

ADD CONSTRAINT reservation\_availiability\_fk FOREIGN KEY ( availiability\_availiability\_id )

REFERENCES availiability ( availiability\_id );

ALTER TABLE reservation

ADD CONSTRAINT reservation\_b\_office\_fk FOREIGN KEY ( b\_office\_b\_office\_id )

REFERENCES b\_office ( b\_office\_id );

ALTER TABLE reservation

ADD CONSTRAINT reservation\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE reservation

ADD CONSTRAINT reservation\_flight\_fk FOREIGN KEY ( flight\_flight\_id )

REFERENCES flight ( flight\_id );

ALTER TABLE reservation

ADD CONSTRAINT reservation\_ticket\_cost\_fk FOREIGN KEY ( ticket\_cost\_ticket\_cost\_id )

REFERENCES ticket\_cost ( ticket\_cost\_id );

ALTER TABLE ticket\_cost

ADD CONSTRAINT ticket\_cost\_reservation\_fk FOREIGN KEY ( reservation\_reservation\_id )

REFERENCES reservation ( reservation\_id );

CREATE SEQUENCE availiability\_availiability\_id START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER availiability\_availiability\_id BEFORE

INSERT ON availiability

FOR EACH ROW

WHEN ( new.availiability\_id IS NULL )

BEGIN

:new.availiability\_id := availiability\_availiability\_id.nextval;

END;

/

CREATE SEQUENCE b\_office\_b\_office\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER b\_office\_b\_office\_id\_trg BEFORE

INSERT ON b\_office

FOR EACH ROW

WHEN ( new.b\_office\_id IS NULL )

BEGIN

:new.b\_office\_id := b\_office\_b\_office\_id\_seq.nextval;

END;

/

CREATE SEQUENCE city\_city\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER city\_city\_id\_trg BEFORE

INSERT ON city

FOR EACH ROW

WHEN ( new.city\_id IS NULL )

BEGIN

:new.city\_id := city\_city\_id\_seq.nextval;

END;

/

CREATE SEQUENCE customer\_customer\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER customer\_customer\_id\_trg BEFORE

INSERT ON customer

FOR EACH ROW

WHEN ( new.customer\_id IS NULL )

BEGIN

:new.customer\_id := customer\_customer\_id\_seq.nextval;

END;

/

CREATE SEQUENCE flight\_flight\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER flight\_flight\_id\_trg BEFORE

INSERT ON flight

FOR EACH ROW

WHEN ( new.flight\_id IS NULL )

BEGIN

:new.flight\_id := flight\_flight\_id\_seq.nextval;

END;

/

CREATE SEQUENCE reservation\_reservation\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER reservation\_reservation\_id\_trg BEFORE

INSERT ON reservation

FOR EACH ROW

WHEN ( new.reservation\_id IS NULL )

BEGIN

:new.reservation\_id := reservation\_reservation\_id\_seq.nextval;

END;

/

CREATE SEQUENCE ticket\_cost\_ticket\_cost\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER ticket\_cost\_ticket\_cost\_id\_trg BEFORE

INSERT ON ticket\_cost

FOR EACH ROW

WHEN ( new.ticket\_cost\_id IS NULL )

BEGIN

:new.ticket\_cost\_id := ticket\_cost\_ticket\_cost\_id\_seq.nextval;

END;

/

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 11

-- CREATE INDEX 12

-- ALTER TABLE 27

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 7

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 7

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

-- TSDP POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 4

-- WARNINGS 0