CREATE TABLE

```
CREATE TABLE classifica
(
  data timestamp without time zone,
  id_hospede integer,
  id_hotel integer,
  mensagem text,
  id_classifica serial NOT NULL,
  controle serial NOT NULL,
  CONSTRAINT classifica_pkey PRIMARY KEY (id_classifica),
  CONSTRAINT classifica_id_hospede_fkey FOREIGN KEY (id_hospede)
      REFERENCES hospede (id_hospede) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE NO ACTION,
  CONSTRAINT classifica_id_hotel_fkey FOREIGN KEY (id_hotel)
      REFERENCES hotel (id_hotel) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE CASCADE
)
```

```
CREATE TABLE consome
(
  qtd integer,
  data date,
  id_prod integer,
  id_reserva integer NOT NULL,
  CONSTRAINT consume_pkey PRIMARY KEY (id_reserva),
  CONSTRAINT consome_id_prod_fkey FOREIGN KEY (id_prod)
      REFERENCES produto (id_prod) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE NO ACTION,
  CONSTRAINT consome_id_reserva_fkey FOREIGN KEY (id_reserva)
      REFERENCES reserva (id_reserva) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE CASCADE
)
```

```
CREATE TABLE estoque
(
  qtd integer,
  id_prod integer,
  id_quarto integer NOT NULL,
  CONSTRAINT estoque_pkey PRIMARY KEY (id_quarto),
  CONSTRAINT estoque_id_prod_fkey FOREIGN KEY (id_prod)
      REFERENCES produto (id_prod) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE NO ACTION,
  CONSTRAINT estoque_id_quarto_fkey FOREIGN KEY (id_quarto)
      REFERENCES quarto (id_quarto) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE CASCADE
)
```

```
CREATE TABLE faz
(
  id_reserva integer NOT NULL,
  id_hospede integer,
  data timestamp without time zone,
  CONSTRAINT faz_pkey PRIMARY KEY (id_reserva),
  CONSTRAINT faz_id_hospede_fkey FOREIGN KEY (id_hospede)
      REFERENCES hospede (id_hospede) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE NO ACTION,
  CONSTRAINT faz_id_reserva_fkey FOREIGN KEY (id_reserva)
      REFERENCES reserva (id_reserva) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE CASCADE
)
```

```
CREATE TABLE hospede
(
  id_hospede serial NOT NULL,
  cpf character(11),
  login character varying NOT NULL,
  senha character varying NOT NULL,
  nome character varying,
  tel integer,
  email character varying,
  ddd smallint,
  CONSTRAINT hospede_pkey PRIMARY KEY (id_hospede),
  CONSTRAINT hospede_cpf_key UNIQUE (cpf),
  CONSTRAINT hospede_login_key UNIQUE (login)
)
```

```
CREATE TABLE hotel
(
  id_hotel serial NOT NULL,
  nome character varying,
  endereco character varying,
  tel integer,
  cidade character varying,
  id_pais integer,
  ddd smallint,
  CONSTRAINT hotel_pkey PRIMARY KEY (id_hotel),
  CONSTRAINT hotel_id_pais_fkey FOREIGN KEY (id_pais)
      REFERENCES pais (id_pais) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
```

```
CREATE TABLE pais
(
  id_pais serial NOT NULL,
  nome character varying,
  capital character varying,
  ddi integer,
  moeda character varying,
  CONSTRAINT pais_pkey PRIMARY KEY (id_pais)
)
```

```
CREATE TABLE produto
(
  id_prod serial NOT NULL,
  valor numeric(6,2),
  nome character varying,
  CONSTRAINT produto_pkey PRIMARY KEY (id_prod)
)
```

```
CREATE TABLE tem
(
  id_reserva integer NOT NULL,
  id_quarto integer,
  CONSTRAINT tem_pkey PRIMARY KEY (id_reserva),
  CONSTRAINT tem_id_quarto_fkey FOREIGN KEY (id_quarto)
      REFERENCES quarto (id_quarto) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT tem_id_reserva_fkey FOREIGN KEY (id_reserva)
      REFERENCES reserva (id_reserva) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE CASCADE
)
```

```
CREATE TABLE quarto
(
  id_quarto serial NOT NULL,
  num_quarto integer NOT NULL,
  andar integer NOT NULL,
  id_tipo integer,
  id_hotel integer,
  CONSTRAINT quarto_pkey PRIMARY KEY (id_quarto),
  CONSTRAINT quarto_id_hotel_fkey FOREIGN KEY (id_hotel)
      REFERENCES hotel (id_hotel) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE CASCADE,
  CONSTRAINT quarto_id_tipo_fkey FOREIGN KEY (id_tipo)
      REFERENCES tipoquarto (id_tipo) MATCH SIMPLE
      ON UPDATE CASCADE ON DELETE SET NULL
)
```

```
CREATE TABLE reserva
(
  id_reserva serial NOT NULL,
  reserva_de character varying,
  sitpg boolean,
  checkin timestamp without time zone,
  checkout timestamp without time zone,
  reserva_ate character varying,
  controle serial NOT NULL,
  CONSTRAINT reserva_pkey PRIMARY KEY (id_reserva)
)
```

```
CREATE TABLE tipoquarto
(
  id_tipo serial NOT NULL,
  qtdcamas integer,
  fumantearea boolean,
  lotacao integer,
  valor numeric(6,2),
  descri text,
  nome character varying,
  CONSTRAINT tipoquarto_pkey PRIMARY KEY (id_tipo)
)
```

TRIGGER

```
CREATE TABLE log_altera_reserva
(
  id serial NOT NULL,
  data timestamp without time zone,
  id_reserva integer,
  de character varying,
  ate character varying,
  CONSTRAINT log_altera_reserva_pkey PRIMARY KEY (id)
)

CREATE OR REPLACE FUNCTION sp_altera_reserva()
  RETURNS trigger AS
$BODY$
begin
	insert into log_altera_reserva(data,id_reserva,de,ate) values
(now(),old.id_reserva,old.reserva_de,old.reserva_ate);
	return null;
end;
$BODY$
  LANGUAGE plpgsql

CREATE TRIGGER tr_altera_reserva
  AFTER UPDATE
  ON reserva
  FOR EACH ROW
  EXECUTE PROCEDURE sp_altera_reserva();
```

```
CREATE TABLE log_cancela_reserva
(
  id serial NOT NULL,
  data timestamp without time zone,
  id_reserva integer,
  de character varying,
  ate character varying,
  num_quarto integer,
  andar_quarto integer,
  hospede integer,
  hotel character varying,
  CONSTRAINT log_cancela_reserva_pkey PRIMARY KEY (id)
)
```

```
CREATE OR REPLACE FUNCTION sp_cancela_reserva()  RETURNS trigger AS
$BODY$
declare
qnum int;
qandar int;
rhospede int;
rde varchar;
rate varchar;
rhotel varchar;
begin
	select hospede,de,ate,hotel,andar,quarto into rhospede,rde,rate,rhotel,qandar,qnum from
viewreserva where codigo=old.id_reserva;
	insert into log_cancela_reserva(data,id_reserva) values(now(),old.id_reserva);
	return old;
end;
$BODY$
  LANGUAGE plpgsql

CREATE TRIGGER tr_cancela_reserva
  AFTER DELETE
  ON reserva
  FOR EACH ROW
  EXECUTE PROCEDURE sp_cancela_reserva();
```

```
CREATE TABLE log_class
(
  id serial NOT NULL,
  id_class integer,
  mesg text,
  data timestamp without time zone,
  CONSTRAINT log_class_pkey PRIMARY KEY (id)
)

CREATE OR REPLACE FUNCTION sp_alter_class()
  RETURNS trigger AS
$BODY$
begin
	insert into log_class(id_class, mesg,data) values(old.id_classifica,old.mensagem,now());
	return null;
end;
$BODY$
  LANGUAGE plpgsql

CREATE TRIGGER td_alter_class
  AFTER UPDATE
  ON classifica
  FOR EACH ROW
  EXECUTE PROCEDURE sp_alter_class();
```

```
CREATE TABLE logdelecao
(
  data timestamp without time zone,
  nome character varying,
  email character varying,
  id serial NOT NULL,
  CONSTRAINT logdelecao_pkey PRIMARY KEY (id)
)

CREATE OR REPLACE FUNCTION sp_logdelecao()
  RETURNS trigger AS
$BODY$
begin
	insert into logDelecao values (now(),old.nome,old.email);
	return old;
end;
$BODY$
  LANGUAGE plpgsql

CREATE TRIGGER tr_logdelecao
  AFTER DELETE
  ON hospede
  FOR EACH ROW
  EXECUTE PROCEDURE sp_logdelecao();
```

VIEW

```
CREATE OR REPLACE VIEW viewhotel AS
 SELECT hotel.id_hotel AS codigo,
    hotel.nome,
    hotel.endereco,
    hotel.ddd,
    hotel.tel,
    pais.nome AS pais,
    hotel.cidade
   FROM hotel,
    pais
  WHERE hotel.id_pais = pais.id_pais;
```

```
CREATE OR REPLACE VIEW viewquartos AS
 SELECT quarto.num_quarto,
    quarto.andar,
    tipoquarto.nome AS tipo,
    tipoquarto.descri,
    hotel.id_hotel AS hotel
   FROM quarto,
    tipoquarto,
    hotel
  WHERE quarto.id_hotel = hotel.id_hotel AND tipoquarto.id_tipo = quarto.id_tipo;
```

```
CREATE OR REPLACE VIEW viewreserva AS
 SELECT hospede.id_hospede AS hospede,
    reserva.id_reserva AS codigo,
    reserva.reserva_de AS de,
    reserva.reserva_ate AS ate,
    reserva.checkin,
    reserva.checkout,
    reserva.sitpg AS pagamento,
    hotel.nome AS hotel,
    quarto.andar,
    quarto.num_quarto AS quarto,
    faz.data,
    tipoquarto.nome AS tipo,
    hotel.id_hotel
   FROM reserva,
    quarto,
    tem,
    faz,
    tipoquarto,
    hospede,
    hotel
  WHERE faz.id_reserva = reserva.id_reserva AND tem.id_reserva = reserva.id_reserva AND
```

```
tem.id_quarto = quarto.id_quarto AND tipoquarto.id_tipo = quarto.id_tipo AND
hospede.id_hospede = faz.id_hospede AND hotel.id_hotel = quarto.id_hotel;
```