

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

APLICACIONES INFORMÁTICAS II

APLICACIÓN WEB: SOLUCIONES METÁLICAS

Huisha Cristina

7107

r object

ob
10

AD 58457-DJ JK

MODELADO DE DATOS

SCRIPT

BEGIN;

CREATE TABLE IF NOT EXISTS public.area

```
(
    id_area serial NOT NULL,
    nombre_area character varying(255) COLLATE pg_catalog."default" NOT NULL,
    descripcion_area character varying(255) COLLATE pg_catalog."default",
    CONSTRAINT area_pkey PRIMARY KEY (id_area)
);
```

CREATE TABLE IF NOT EXISTS public.asignacion

```
(
    id_contrato integer NOT NULL,
    id_area integer NOT NULL,
    id_empleado integer NOT NULL,
    CONSTRAINT asignacion_pkey PRIMARY KEY (id_contrato, id_area, id_empleado)
);
```

CREATE TABLE IF NOT EXISTS public.contrato

```
(
    id_contrato serial NOT NULL,
    cliente character varying(100) COLLATE pg_catalog."default" NOT NULL,
    fecha_inicio date NOT NULL,
    fecha_fin date NOT NULL,
    presupuesto numeric(10, 2) NOT NULL,
    id_estado integer NOT NULL,
    CONSTRAINT contrato_pkey PRIMARY KEY (id_contrato)
);
```

);

CREATE TABLE IF NOT EXISTS public.desempeno

(

id_desempeno serial NOT NULL,

"puntuacion_desempeño" character varying(25) COLLATE pg_catalog."default" NOT NULL,

CONSTRAINT desempeno_pkey PRIMARY KEY (id_desempeno)

);

CREATE TABLE IF NOT EXISTS public.empleado

(

id_empleado serial NOT NULL,

nombres character varying(255) COLLATE pg_catalog."default" NOT NULL,

apellidos character varying(255) COLLATE pg_catalog."default" NOT NULL,

cedula character varying(10) COLLATE pg_catalog."default" NOT NULL,

tiempo_exp_general integer NOT NULL,

numero_contacto character varying(10) COLLATE pg_catalog."default" NOT NULL,

CONSTRAINT empleado_pkey PRIMARY KEY (id_empleado)

);

CREATE TABLE IF NOT EXISTS public.estado

(

id_estado serial NOT NULL,

estado character varying(20) COLLATE pg_catalog."default" NOT NULL,

CONSTRAINT estado_pkey PRIMARY KEY (id_estado)

);

```
CREATE TABLE IF NOT EXISTS public.estadoarea
```

```
(  
    id_contrato integer NOT NULL,  
    id_area integer NOT NULL,  
    id_estado integer NOT NULL,  
    id_proyecto integer NOT NULL,  
    CONSTRAINT estadoarea_pkey PRIMARY KEY (id_contrato, id_area, id_estado,  
id_proyecto)  
);
```

```
CREATE TABLE IF NOT EXISTS public.evaluacion
```

```
(  
    id_desempeno integer NOT NULL,  
    id_area integer NOT NULL,  
    id_empleado integer NOT NULL,  
    CONSTRAINT evaluacion_pkey PRIMARY KEY (id_desempeno, id_area,  
id_empleado)  
);
```

```
CREATE TABLE IF NOT EXISTS public.presupuesto
```

```
(  
    id_contrato integer NOT NULL,  
    id_area integer NOT NULL,  
    id_proyecto integer NOT NULL,  
    monto numeric(10, 2) NOT NULL,  
    CONSTRAINT presupuesto_pkey PRIMARY KEY (id_contrato, id_area,  
id_proyecto)  
);
```

```
CREATE TABLE IF NOT EXISTS public.producto
```

```
(  
    id_producto serial NOT NULL,  
    nombre_producto character varying(100) COLLATE pg_catalog."default" NOT  
NULL,  
    descripcion_producto character varying(255) COLLATE pg_catalog."default",  
    CONSTRAINT producto_pkey PRIMARY KEY (id_producto)  
);
```

CREATE TABLE IF NOT EXISTS public.proyecto

```
(  
    id_proyecto serial NOT NULL,  
    id_contrato integer NOT NULL,  
    id_producto integer NOT NULL,  
    cantidad_producto integer NOT NULL,  
    fecha_inicio date NOT NULL,  
    fecha_fin date NOT NULL,  
    id_estado integer NOT NULL,  
    CONSTRAINT proyecto_pkey PRIMARY KEY (id_proyecto)  
);
```

CREATE TABLE IF NOT EXISTS public.recomendacion

```
(  
    id_contrato integer NOT NULL,  
    id_proyecto integer NOT NULL,  
    id_area integer NOT NULL,  
    recomendacion_num_e integer NOT NULL,  
    cantidad_asignada integer NOT NULL,  
    CONSTRAINT recomendacion_pkey PRIMARY KEY (id_contrato, id_proyecto,  
id_area)
```

);

CREATE TABLE IF NOT EXISTS public.reglas

```
(
    id_reglas serial NOT NULL,
    nombre character varying(255) COLLATE pg_catalog."default" NOT NULL,
    valor integer,
    CONSTRAINT reglas_pkey PRIMARY KEY (id_reglas)
);
```

CREATE TABLE IF NOT EXISTS public.rol

```
(
    id_rol serial NOT NULL,
    nombre_rol character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT rol_pkey PRIMARY KEY (id_rol)
);
```

CREATE TABLE IF NOT EXISTS public.usuario

```
(
    nombre character varying(200) COLLATE pg_catalog."default" NOT NULL,
    correo_electronico character varying(255) COLLATE pg_catalog."default" NOT NULL,
    contrasena character varying(255) COLLATE pg_catalog."default" NOT NULL,
    id_rol integer NOT NULL,
    CONSTRAINT usuario_pkey PRIMARY KEY (nombre)
);
```

ALTER TABLE IF EXISTS public.asignacion

ADD CONSTRAINT asignacion_id_area_fkey FOREIGN KEY (id_area)

REFERENCES public.area (id_area) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.asignacion

ADD CONSTRAINT asignacion_id_contrato_fkey FOREIGN KEY (id_contrato)

REFERENCES public.contrato (id_contrato) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.asignacion

ADD CONSTRAINT asignacion_id_empleado_fkey FOREIGN KEY (id_empleado)

REFERENCES public.empleado (id_empleado) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.contrato

ADD CONSTRAINT contrato_id_estado_fkey FOREIGN KEY (id_estado)

REFERENCES public.estado (id_estado) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.estadoarea

ADD CONSTRAINT estadoarea_id_area_fkey FOREIGN KEY (id_area)

REFERENCES public.area (id_area) MATCH SIMPLE

ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.estadoarea
ADD CONSTRAINT estadoarea_id_contrato_fkey FOREIGN KEY (id_contrato)
REFERENCES public.contrato (id_contrato) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.estadoarea
ADD CONSTRAINT estadoarea_id_estado_fkey FOREIGN KEY (id_estado)
REFERENCES public.estado (id_estado) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.estadoarea
ADD CONSTRAINT estadoarea_id_proyecto_fkey FOREIGN KEY (id_proyecto)
REFERENCES public.proyecto (id_proyecto) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.evaluacion
ADD CONSTRAINT evaluacion_id_area_fkey FOREIGN KEY (id_area)
REFERENCES public.area (id_area) MATCH SIMPLE
ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.evaluacion

ADD CONSTRAINT evaluacion_id_empleado_fkey FOREIGN KEY (id_empleado)

REFERENCES public.empleado (id_empleado) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.evaluacion

ADD CONSTRAINT evaluacion_id_desempeno_fkey FOREIGN KEY
(id_desempeno)

REFERENCES public.desempeno (id_desempeno) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.presupuesto

ADD CONSTRAINT presupuesto_id_area_fkey FOREIGN KEY (id_area)

REFERENCES public.area (id_area) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.presupuesto

ADD CONSTRAINT presupuesto_id_contrato_fkey FOREIGN KEY (id_contrato)

REFERENCES public.contrato (id_contrato) MATCH SIMPLE

ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.presupuesto
ADD CONSTRAINT presupuesto_id_proyecto_fkey FOREIGN KEY (id_proyecto)
REFERENCES public.proyecto (id_proyecto) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.proyecto
ADD CONSTRAINT proyecto_id_contrato_fkey FOREIGN KEY (id_contrato)
REFERENCES public.contrato (id_contrato) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.proyecto
ADD CONSTRAINT proyecto_id_estado_fkey FOREIGN KEY (id_estado)
REFERENCES public.estado (id_estado) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.proyecto
ADD CONSTRAINT proyecto_id_producto_fkey FOREIGN KEY (id_producto)
REFERENCES public.producto (id_producto) MATCH SIMPLE
ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.recomendacion

ADD CONSTRAINT recomendacion_id_area_fkey FOREIGN KEY (id_area)

REFERENCES public.area (id_area) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.recomendacion

ADD CONSTRAINT recomendacion_id_contrato_fkey FOREIGN KEY (id_contrato)

REFERENCES public.contrato (id_contrato) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.recomendacion

ADD CONSTRAINT recomendacion_id_proyecto_fkey FOREIGN KEY (id_proyecto)

REFERENCES public.proyecto (id_proyecto) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.usuario

ADD CONSTRAINT usuario_id_rol_fkey FOREIGN KEY (id_rol)

REFERENCES public.rol (id_rol) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

END;

Tablas y Relaciones

1. Tabla area

- **id_area**: Identificador único de la área (Primary Key).
- **nombre_area**: Nombre del área (No Nulo).
- **descripcion_area**: Descripción del área (Opcional).

2. Tabla asignacion

- **id_contrato**: Identificador del contrato (Parte de la Primary Key).
- **id_area**: Identificador del área (Parte de la Primary Key).
- **id_empleado**: Identificador del empleado asignado (Parte de la Primary Key).

3. Tabla contrato

- **id_contrato**: Identificador único del contrato (Primary Key).
- **cliente**: Nombre del cliente (No Nulo).
- **fecha_inicio**: Fecha de inicio del contrato (No Nulo).
- **fecha_fin**: Fecha de finalización del contrato (No Nulo).
- **presupuesto**: Presupuesto asignado al contrato (No Nulo).
- **id_estado**: Identificador del estado del contrato (No Nulo).

4. Tabla desempeno

- **id_desempeno**: Identificador único del desempeño (Primary Key).
- **puntuacion_desempeno**: Puntuación del desempeño (No Nulo).

5. Tabla empleado

- **id_empleado**: Identificador único del empleado (Primary Key).
- **nombres**: Nombres del empleado (No Nulo).
- **apellidos**: Apellidos del empleado (No Nulo).
- **cedula**: Número de cédula del empleado (No Nulo).

- **tiempo_exp_general:** Tiempo de experiencia general del empleado en años (No Nulo).
- **numero_contacto:** Número de contacto del empleado (No Nulo).

6. Tabla estado

- **id_estado:** Identificador único del estado (Primary Key).
- **estado:** Descripción del estado (No Nulo).

7. Tabla producto

- **id_producto:** Identificador único del producto (Primary Key).
- **nombre_producto:** Nombre del producto (No Nulo).
- **descripcion_producto:** Descripción del producto (Opcional).

8. Tabla proyecto

- **id_proyecto:** Identificador único del proyecto (Primary Key).
- **id_contrato:** Identificador del contrato asociado (No Nulo, Foreign Key).
- **id_estado:** Identificador del estado del proyecto (No Nulo, Foreign Key).
- **id_producto:** Identificador del producto asociado (No Nulo, Foreign Key).
- **fecha_inicio:** Fecha de inicio del proyecto (No Nulo).
- **fecha_fin:** Fecha de finalización del proyecto (No Nulo).

9. Tabla presupuesto

- **id_presupuesto:** Identificador único del presupuesto (Primary Key).
- **id_contrato:** Identificador del contrato asociado (No Nulo, Foreign Key).
- **id_proyecto:** Identificador del proyecto asociado (No Nulo, Foreign Key).
- **monto:** Monto asignado al presupuesto (No Nulo).

10. Tabla recomendacion

- **id_recomendacion:** Identificador único de la recomendación (Primary Key).
- **id_area:** Identificador del área asociada (No Nulo, Foreign Key).

- **id_contrato**: Identificador del contrato asociado (No Nulo, Foreign Key).
- **id_proyecto**: Identificador del proyecto asociado (No Nulo, Foreign Key).
- **descripcion_recomendacion**: Descripción de la recomendación (Opcional).

11. Tabla rol

- **id_rol**: Identificador único del rol (Primary Key).
- **nombre_rol**: Nombre del rol (No Nulo).

12. Tabla usuario

- **id_usuario**: Identificador único del usuario (Primary Key).
- **nombre_usuario**: Nombre de usuario (No Nulo).
- **contrasena**: Contraseña del usuario (No Nulo).
- **id_rol**: Identificador del rol del usuario (No Nulo, Foreign Key).

Relaciones entre Tablas

- **proyecto** tiene una relación muchos-a-uno con **contrato**, **estado**, y **producto**.
- **presupuesto** tiene una relación muchos-a-uno con **contrato** y **proyecto**.
- **asignacion** tiene una relación muchos-a-muchos con **contrato**, **area**, y **empleado**.
- **recomendacion** tiene una relación muchos-a-uno con **area**, **contrato**, y **proyecto**.
- **usuario** tiene una relación muchos-a-uno con **rol**.

Estas tablas y relaciones permiten gestionar los contratos, proyectos, áreas, presupuestos, recomendaciones, empleados, sus desempeños, y los usuarios con sus roles dentro de la empresa de Soluciones Metálicas.