

CRUD (Create, Read, Update, Delete) em PostgreSQL para as entidades do Banco de dados.

Lembre-se de adaptar esses comandos conforme necessário. Você pode adicionar cláusulas adicionais, como condições WHERE, para refinar as operações CRUD de acordo com suas necessidades.

Tabela “country”

1. Create (Criar um novo registro):

```
INSERT INTO "country" ("name") VALUES ('Brasil');
```

Este comando insere um novo registro na tabela “country” com o valor ‘Brasil’ na coluna “name”. O valor para a coluna “country_id” será gerado automaticamente, devido à configuração da coluna como SERIAL.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "country";
```

Este comando retorna todos os registros da tabela “country”.

```
SELECT * FROM "country" WHERE "country_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "country" SET "name" = 'Uruguai' WHERE "country_id" = 1;
```

Este comando atualiza o valor da coluna “name” para ‘Uruguai’ no registro com “country_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "country" WHERE "country_id" = 1;
```

Este comando exclui o registro com “country_id” igual a 1 da tabela “country”.

Tabela “state”

1. Create (Criar um novo registro):

```
INSERT INTO "state" ("country_id", "name") VALUES (1, 'Rio de Janeiro');
```

Este comando insere um novo registro na tabela “state” com o valor 1 na coluna “country_id” (referenciando o país) e ‘Rio de Janeiro’ na coluna “name”. O valor para a coluna “state_id” será gerado automaticamente devido à configuração da coluna como SERIAL.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "state";
```

Este comando retorna todos os registros da tabela “state”.

```
SELECT * FROM "state" WHERE "state_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "state" SET "name" = 'São Paulo' WHERE "state_id" = 1;
```

Este comando atualiza o valor da coluna “name” para ‘São Paulo’ no registro com “state_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "state" WHERE "state_id" = 1;
```

Este comando exclui o registro com “state_id” igual a 1 da tabela “state”.

Tabela “city”

1. Create (Criar um novo registro):

```
INSERT INTO "city" ("state_id", "name") VALUES (1, 'São Paulo');
```

Este comando insere um novo registro na tabela “city” com o valor 1 na coluna “state_id” (referenciando o estado) e ‘São Paulo’ na coluna “name”. O valor para a coluna “city_id” será gerado automaticamente devido à configuração da coluna como SERIAL.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "city";
```

Este comando retorna todos os registros da tabela “city”.

```
SELECT * FROM "city" WHERE "city_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "city" SET "name" = 'Rio de Janeiro' WHERE "city_id" = 1;
```

Este comando atualiza o valor da coluna “name” para ‘Rio de Janeiro’ no registro com “city_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "city" WHERE "city_id" = 1;
```

Este comando exclui o registro com “city_id” igual a 1 da tabela “city”.

Tabela “user_register”

1. Create (Criar um novo registro):

```
INSERT INTO "user_register" (  
  "name", "last_name", "email", "cpf", "rg", "password", "birth_date",  
  "city_id", "address", "number", "address_complement", "postal_code",  
  "phone", "mobile_phone"  
) VALUES (  
  'Usuario', 'Teste', 'teste@exemplo.com', '1234567890', '987654321', 'minhasenha', '1990-01-01',  
  1, 'Rua Santa Filomena', '123', 'Apt 4', '12345-678', '123-456-7890', '987-654-3210'  
);
```

Este comando insere um novo registro na tabela “user_register” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "user_register";
```

Este comando retorna todos os registros da tabela “user_register”.

```
SELECT * FROM "user_register" WHERE "user_register_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "user_register" SET "name" = 'Novo' WHERE "user_id" = 1;
```

Este comando atualiza o valor da coluna “name” para ‘Jane’ no registro com “user_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "user_register" WHERE "user_id" = 1;
```

Este comando exclui o registro com “user_id” igual a 1 da tabela “user_register”.

Tabela “familiar”

1. Create (Criar um novo registro):

```
INSERT INTO "familiar" (  
  "user_id", "name", "kinship", "email", "phone", "mobile_phone"  
) VALUES (  
  1, 'Fulano de tal', 'Pai', 'contato@exemplo.com', '123-456-7890', '987-654-3210'  
);
```

Este comando insere um novo registro na tabela “familiar” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "familiar";
```

Este comando retorna todos os registros da tabela “familiar”.

```
SELECT * FROM "familiar" WHERE "familiar_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "familiar" SET "phone" = '555-555-5555' WHERE "familiar_id" = 1;
```

Este comando atualiza o valor da coluna “phone” para ‘555-555-5555’ no registro com “familiar_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "familiar" WHERE "familiar_id" = 1;
```

Este comando exclui o registro com “familiar_id” igual a 1 da tabela “familiar”.

Tabela “social”

1. Create (Criar um novo registro):

```
INSERT INTO "social" (  
  "user_id", "type_music", "favorite_sport", "favorite_team", "favorite_card_game",  
  "favorite_board_game", "favorite_news", "favorite_animal_species", "favorite_animal_breed",  
  "facebook_link", "instagram_link"  
) VALUES (  
  1, 'Forró', 'Futebol', 'Botafogo', 'Buraco',  
  'Xadrez', 'Ciências', 'Galinha', 'Embrapa 051', 'facebook.com/user1', 'instagram.com/user1'  
);
```

Este comando insere um novo registro na tabela “social” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "social";
```

Este comando retorna todos os registros da tabela “social”.

```
SELECT * FROM "social" WHERE "social_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "social" SET "type_music" = 'Pop' WHERE "social_id" = 1;
```

Este comando atualiza o valor da coluna “type_music” para ‘Pop’ no registro com “social_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "social" WHERE "social_id" = 1;
```

Este comando exclui o registro com “social_id” igual a 1 da tabela “social”.

Tabela “health”

1. Create (Criar um novo registro):

```
INSERT INTO "health" (  
  "user_id", "height", "weight", "allergies", "medical_insurance",  
  "medical_insurance_register", "hospital_for_removal", "private_physician_name",  
  "private_physician_phone", "private_physician_email"  
) VALUES (  
  1, , 1.75, 70.5, 'Abelha', 'Amil', '12345', 'Hospital X',  
  'Dr. Drauzio', '123-456-7890', 'drauzio@exemplo.com'  
);
```

Este comando insere um novo registro na tabela “health” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "health";
```

Este comando retorna todos os registros da tabela “health”.

```
SELECT * FROM "health" WHERE "health_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "health" SET "height" = 1.80 WHERE "health_id" = 1;
```

Este comando atualiza o valor da coluna “height” para 1.80 no registro com “health_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "health" WHERE "health_id" = 1;
```

Este comando exclui o registro com “health_id” igual a 1 da tabela “health”.

Tabela “medicine”

1. Create (Criar um novo registro):

```
INSERT INTO "medicine" (  
  "user_id", "medicine_name", "medicine_dosage"  
) VALUES (  
  1, 'Remédio A', '1 comprimido ao dia'  
);
```

Este comando insere um novo registro na tabela “medicine” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "medicine";
```

Este comando retorna todos os registros da tabela “medicine”.

```
SELECT * FROM "medicine" WHERE "medicine_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "medicine" SET "medicine_dosage" = '2 pills per day' WHERE "medicine_id" = 1;
```

Este comando atualiza o valor da coluna “medicine_dosage” para ‘2 pills per day’ no registro com “medicine_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "medicine" WHERE "medicine_id" = 1;
```

Este comando exclui o registro com “medicine_id” igual a 1 da tabela “medicine”.

Tabela “diet”

1. Create (Criar um novo registro):

```
INSERT INTO "diet" (  
  "user_id", "favorite_food"  
) VALUES (  
  1, 'Pizza'  
);
```

Este comando insere um novo registro na tabela “diet” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "diet";
```

Este comando retorna todos os registros da tabela “diet”.

```
SELECT * FROM "diet" WHERE "diet_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "diet" SET "favorite_food" = 'Sushi' WHERE "diet_id" = 1;
```

Este comando atualiza o valor da coluna “favorite_food” para ‘Sushi’ no registro com “diet_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "diet" WHERE "diet_id" = 1;
```

Este comando exclui o registro com “diet_id” igual a 1 da tabela “diet”.

Tabela “medical_history”

1. Create (Criar um novo registro):

```
INSERT INTO "medical_history" (  
    "user_id", "medical_history_resume"  
) VALUES (  
    1, 'Resumo do histórico médico do usuário'  
);
```

Este comando insere um novo registro na tabela “medical_history” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "medical_history";
```

Este comando retorna todos os registros da tabela “medical_history”.

```
SELECT * FROM "medical_history" WHERE "medical_history_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "medical_history" SET "medical_history_resume" = 'Novo resumo do histórico médico' WHERE "medical_history_id" = 1;
```

Este comando atualiza o valor da coluna “medical_history_resume” para ‘Novo resumo do histórico médico’ no registro com “medical_history_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "medical_history" WHERE "medical_history_id" = 1;
```

Este comando exclui o registro com “medical_history_id” igual a 1 da tabela “medical_history”.

Tabela “medical_exams”

1. Create (Criar um novo registro):

```
INSERT INTO "medical_exams" (  
    "user_id", "ecg", "pa", "sat_o2", "glycemia", "other_exams"  
) VALUES (  
    1, 'caminho/para/imagem/ECG', 'caminho/para/imagem/PA', 'Saturação de O2', 'Glicemia', 'Outros exames'  
);
```

Este comando insere um novo registro na tabela “medical_exams” com os valores fornecidos. Neste ponto é importante frisar que podem ser inseridas informações textuais ou o endereço da imagem no servidor.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "medical_exams";
```

Este comando retorna todos os registros da tabela “medical_exams”.

```
SELECT * FROM "medical_exams" WHERE "medical_exams_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "medical_exams" SET "glycemia" = 'Novo valor de glicemia' WHERE "medical_exams_id" =
```

Este comando atualiza o valor da coluna “glycemia” para ‘Novo valor de glicemia’ no registro com “medical_exams_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "medical_exams" WHERE "medical_exams_id" = 1;
```

Este comando exclui o registro com “medical_exams_id” igual a 1 da tabela “medical_exams”.

Tabela “plan__type”

1. Create (Criar um novo registro):

```
INSERT INTO "plan_type" (  
  "plan_type", "plan_resume"  
) VALUES (  
  'Tipo de Plano A', 'Resumo do plano A'  
);
```

Este comando insere um novo registro na tabela “plan__type” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "plan_type";
```

Este comando retorna todos os registros da tabela “plan__type”.

```
SELECT * FROM "plan_type" WHERE "plan_type_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "plan_type" SET "plan_resume" = 'Novo resumo do plano A' WHERE "plan_type_id" = 1;
```

Este comando atualiza o valor da coluna “plan__resume” para ‘Novo resumo do plano A’ no registro com “plan__type_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "plan_type" WHERE "plan_type_id" = 1;
```


Este comando exclui o registro com “plan_type_id” igual a 1 da tabela “plan_type”.

Tabela “company”

1. Create (Criar um novo registro):

```
INSERT INTO "company" (  
  "name", "cnpj", "address", "number", "city_id", "address_complement",  
  "phone", "email", "website", "company_resume"  
) VALUES (  
  'Nome da Empresa', '1234567890', 'Endereço da Empresa', '123', 1, 'Complemento do Endereço',  
  '123-456-7890', 'empresa@example.com', 'www.empresa.com', 'Resumo da Empresa'  
);
```

Este comando insere um novo registro na tabela “company” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "company";
```

Este comando retorna todos os registros da tabela “company”.

```
SELECT * FROM "company" WHERE "company_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "company" SET "phone" = '987-654-3210' WHERE "empresa_id" = 1;
```

Este comando atualiza o valor da coluna “phone” para ‘987-654-3210’ no registro com “empresa_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "company" WHERE "empresa_id" = 1;
```

Este comando exclui o registro com “empresa_id” igual a 1 da tabela “company”.

Tabela “condominium_type”

1. Create (Criar um novo registro):

```
INSERT INTO "condominium_type" (  
  "condominium_type"  
) VALUES (  
  'Tipo de Condomínio A'  
);
```

Este comando insere um novo registro na tabela “condominium_type” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "condominium_type";
```

Este comando retorna todos os registros da tabela “condominium_type”.

```
SELECT * FROM "condominium_type" WHERE "condominium_type_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "condominium_type" SET "condominium_type" = 'Novo Tipo de Condomínio' WHERE "condominium_type_id" = 1;
```

Este comando atualiza o valor da coluna “condominium_type” para ‘Novo Tipo de Condomínio’ no registro com “condominium_type_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "condominium_type" WHERE "condominium_type_id" = 1;
```

Este comando exclui o registro com “condominium_type_id” igual a 1 da tabela “condominium_type”.

Tabela “condominium”

1. Create (Criar um novo registro):

```
INSERT INTO "condominium" (
  "condominium_type", "name", "cnpj", "address", "number", "city_id",
  "address_complement", "phone", "email", "website"
) VALUES (
  1, 'Nome do Condomínio', '1234567890', 'Endereço do Condomínio', '123', 1,
  'Complemento do Endereço', '123-456-7890', 'condominio@example.com', 'www.condominio.com'
);
```

Este comando insere um novo registro na tabela “condominium” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "condominium";
```

Este comando retorna todos os registros da tabela “condominium”.

```
SELECT * FROM "condominium" WHERE "condominium_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "condominium" SET "phone" = '987-654-3210' WHERE "condominium_id" = 1;
```

Este comando atualiza o valor da coluna “phone” para ‘987-654-3210’ no registro com “condominium_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "condominium" WHERE "condominium_id" = 1;
```

Este comando exclui o registro com “condominium_id” igual a 1 da tabela “condominium”.

Tabela “property_type”

1. Create (Criar um novo registro):

```
INSERT INTO "property_type" (  
  "property_type"  
) VALUES (  
  'Tipo de Propriedade A'  
)  
);
```

Este comando insere um novo registro na tabela “property_type” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "property_type";
```

Este comando retorna todos os registros da tabela “property_type”.

```
SELECT * FROM "property_type" WHERE "property_type_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "property_type" SET "property_type" = 'Novo Tipo de Propriedade' WHERE "property_type_id" = 1;
```

Este comando atualiza o valor da coluna “property_type” para ‘Novo Tipo de Propriedade’ no registro com “property_type_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "property_type" WHERE "property_type_id" = 1;
```

Este comando exclui o registro com “property_type_id” igual a 1 da tabela “property_type”.

Tabela “property”

1. Create (Criar um novo registro):

```
INSERT INTO "property" (  
  "property_type", "condominium_id", "address", "number", "city_id",  
  "address_complement", "area", "rooms", "bathrooms", "parking_spaces", "floor"  
) VALUES (  
  1, 1, 'Endereço da Propriedade', '123', 1, 'Complemento do Endereço',
```

```
100.5, 3, 2, 2, 5
);
```

Este comando insere um novo registro na tabela “property” com os valores fornecidos.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "property";
```

Este comando retorna todos os registros da tabela “property”.

```
SELECT * FROM "property" WHERE "property_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "property" SET "rooms" = 4 WHERE "property_id" = 1;
```

Este comando atualiza o valor da coluna “rooms” para 4 no registro com “property_id” igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "property" WHERE "property_id" = 1;
```

Este comando exclui o registro com “property_id” igual a 1 da tabela “property”.

Tabela “smart_city”

1. Create (Criar um novo registro):

```
INSERT INTO "smart_city" (
  "plan_type_id"
) VALUES (
  1
);
```

Este comando insere um novo registro na tabela “smart_city” com o valor fornecido.

2. Read (Recuperar registros existentes):

```
SELECT * FROM "smart_city";
```

Este comando retorna todos os registros da tabela “smart_city”.

```
SELECT * FROM "smart_city" WHERE "smart_city_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. Update (Atualizar um registro existente):

```
UPDATE "smart_city" SET "plan_type_id" = 2 WHERE "smart_city_id" = 1;
```

Este comando atualiza o valor da coluna “plan_type_id” para 2 no registro com “smart_city_id” é igual a 1.

4. Delete (Excluir um registro existente):

```
DELETE FROM "smart_city" WHERE "smart_city_id" = 1;
```

Este comando exclui o registro com “smart_city_id” igual a 1 da tabela “smart_city”.

Tabela “affiliation”

1. CREATE (Criar):

```
INSERT INTO "affiliation" ("user_id", "company_id", "condominium_id")  
VALUES (1, 1, 1);
```

Este comando insere um novo registro na tabela “affiliation” com o valor fornecido. Aqui é possível informar se um usuário esta cadastrado a um condomínio, empresa ou ambos.

2. READ (Ler):

```
SELECT * FROM "affiliation";
```

Este comando retorna todos os registros da tabela “affiliation”.

```
SELECT * FROM "affiliation" WHERE "affiliation_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. UPDATE (Atualizar):

```
UPDATE "affiliation"  
SET "user_id" = 2, "company_id" = 2, "condominium_id" = 2  
WHERE "affiliation_id" = 1;
```

Este comando atualiza os valores da tabela “affiliation” onde “affiliation_id” é igual a 1.

4. DELETE (Excluir):

```
DELETE FROM "affiliation" WHERE "affiliation_id" = 1;
```

Este comando exclui o registro com “affiliation_id” igual a 1 da tabela “affiliation”.

Tabela “available_routes”

1. CREATE (Criar):

```
INSERT INTO "available_routes" ("name", "address", "number", "city_id", "address_complement")  
VALUES ('Loja 1', 'Endereço 1', '123', 1, 'Complemento 1', 41.123456, -8.654321, '123-456-78');
```

Este comando insere um novo registro na tabela “available_routes” com os valores fornecidos. Aqui é possível informar lugares cadastrados, com seu endereço e dados de latitude e longitude para cálculo de rotas.

2. READ (Ler):

```
SELECT * FROM "available_routes";
```

Este comando retorna todos os registros da tabela “available_routes”.

```
SELECT * FROM "available_routes" WHERE "available_routes_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. UPDATE (Atualizar):

```
UPDATE "available_routes"
SET "name" = 'Novo Lugar', "address" = 'Endereço Rota Atualizada', "number" = '456',
    "city_id" = 2, "address_complement" = 'Complemento Rota Atualizada', "latitude" = 42.654,
    "longitude" = -9.123456, "phone" = '987-654-3210', "email" = 'rotaatualizada@example.com'
WHERE "available_routes_id" = 1;
```

Este comando atualiza os valores da tabela “available_routes” onde “available_routes_id” é igual a 1.

4. DELETE (Excluir):

```
DELETE FROM "available_routes" WHERE "available_routes_id" = 1;
```

Este comando exclui o registro com “available_routes_id” igual a 1 da tabela “available_routes”.

Tabela “usage_logs”

1. CREATE (Criar):

```
INSERT INTO "usage_logs" ("user_id", "available_routes_id", "notes")
VALUES (1, 1, 'Notas sobre o uso da rota');
```

Este comando insere um novo registro na tabela “usage_logs” com os valores fornecidos. Aqui é possível informar registros de rotas usadas pelo usuário e acrescentar quaisquer informações adicionais.

2. READ (Ler):

```
SELECT * FROM "usage_logs";
```

Este comando retorna todos os registros da tabela “usage_logs”.

```
SELECT * FROM "usage_logs" WHERE "usage_logs_id" = 1;
```

Este comando retorna um registro específico pelo ID

3. UPDATE (Atualizar):

```
UPDATE "usage_logs"
SET "user_id" = 2, "available_routes_id" = 2, "notes" = 'Notas atualizadas sobre o uso da ro
WHERE "usage_logs_id" = 1;
```

Este comando atualiza os valores da tabela “usage_logs” onde “usage_logs_id” é igual a 1.

4. DELETE (Excluir):

```
DELETE FROM "usage_logs" WHERE "usage_logs_id" = 1;
```

Este comando exclui o registro com “usage_logs_id” igual a 1 da tabela “usage_logs”.

Informações de usuário

Todas as informações de usuário

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, f.name AS familiar_name, f.kinship, f.email AS
familiar_email, f.phone AS familiar_phone, so.type_music, so.favorite_sport,
so.favorite_team, so.favorite_card_game, so.favorite_board_game, so.favorite_news,
so.favorite_animal_species, so.favorite_animal_breed, so.facebook_link,
so.instagram_link, d.favorite_food, h.height, h.weight, h.allergies, h.medical_insurance,
h.medical_insurance_register, h.hospital_for_removal, h.private_physician_name,
h.private_physician_phone, h.private_physician_email, m.medicine_name,
m.medicine_dosage, mh.medical_history_resume, me.ecg, me.pa, me.sat_o2,
me.glycemia, me.other_exams FROM “user_register” AS u Left JOIN “city”
AS ci ON u.”city_id” = ci.”city_id” Left JOIN “state” AS st ON ci.”state_id” =
st.”state_id” Left JOIN “country” AS co ON st.”country_id” = co.”country_id”
Left JOIN “familiar” AS f ON u.”user_id” = f.”user_id” Left JOIN “social” AS
so ON u.”user_id” = so.”user_id” Left JOIN “diet” AS d ON u.”user_id” =
d.”user_id” Left JOIN “health” AS h ON u.”user_id” = h.”user_id” Left JOIN
“medicine” AS m ON u.”user_id” = m.”user_id” Left JOIN “medical_history”
AS mh ON u.”user_id” = mh.”user_id” Left JOIN “medical_exams” AS me
ON u.”user_id” = me.”user_id”;
```

Esta consulta retorna todas as informações de usuários agrupados em uma única tabela.

Todas as informações de usuário por usuário

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, f.name AS familiar_name, f.kinship, f.email AS
familiar_email, f.phone AS familiar_phone, so.type_music, so.favorite_sport,
so.favorite_team, so.favorite_card_game, so.favorite_board_game, so.favorite_news,
so.favorite_animal_species, so.favorite_animal_breed, so.facebook_link,
so.instagram_link, d.favorite_food, h.height, h.weight, h.allergies, h.medical_insurance,
h.medical_insurance_register, h.hospital_for_removal, h.private_physician_name,
h.private_physician_phone, h.private_physician_email, m.medicine_name,
m.medicine_dosage, mh.medical_history_resume, me.ecg, me.pa, me.sat_o2,
me.glycemia, me.other_exams FROM "user_register" AS u Left JOIN "city"
AS ci ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
Left JOIN "familiar" AS f ON u."user_id" = f."user_id" Left JOIN "social" AS
so ON u."user_id" = so."user_id" Left JOIN "diet" AS d ON u."user_id" =
d."user_id" Left JOIN "health" AS h ON u."user_id" = h."user_id" Left JOIN
"medicine" AS m ON u."user_id" = m."user_id" Left JOIN "medical_history"
AS mh ON u."user_id" = mh."user_id" Left JOIN "medical_exams" AS me
ON u."user_id" = me."user_id" WHERE u.user_id=1;
```

Esta consulta retorna todas as informações de um único usuário agrupados em uma única tabela.

Dados pessoais

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name,d.favorite_food FROM "user_register" AS u Left
JOIN "city" AS ci ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON
ci."state_id" = st."state_id" Left JOIN "country" AS co ON st."country_id"
= co."country_id" Left JOIN "diet" AS d ON u."user_id" = d."user_id";
```

Esta consulta retorna dados pessoais dos usuários agrupados em uma única tabela.

Dados pessoais por usuário

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name,d.favorite_food FROM "user_register" AS u Left
JOIN "city" AS ci ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON
ci."state_id" = st."state_id" Left JOIN "country" AS co ON st."country_id"
= co."country_id" Left JOIN "diet" AS d ON u."user_id" = d."user_id"
WHERE u.user_id=1;
```

Esta consulta retorna dados pessoais de um único usuário agrupados em uma única tabela.

Dados sociais

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, f.name AS familiar_name, f.kinship, f.email AS
familiar_email, f.phone AS familiar_phone, so.type_music, so.favorite_sport,
so.favorite_team, so.favorite_card_game, so.favorite_board_game, so.favorite_news,
so.favorite_animal_species, so.favorite_animal_breed, so.facebook_link,
so.instagram_link FROM "user_register" AS u Left JOIN "city" AS ci
ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
Left JOIN "familiar" AS f ON u."user_id" = f."user_id" Left JOIN "social"
AS so ON u."user_id" = so."user_id";
```

Esta consulta retorna dados sociais dos usuários agrupados em uma única tabela.

Dados sociais por usuário

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, f.name AS familiar_name, f.kinship, f.email AS
familiar_email, f.phone AS familiar_phone, so.type_music, so.favorite_sport,
```

```
so.favorite_team, so.favorite_card_game, so.favorite_board_game, so.favorite_news,
so.favorite_animal_species, so.favorite_animal_breed, so.facebook_link,
so.instagram_link FROM "user_register" AS u Left JOIN "city" AS ci
ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
Left JOIN "familiar" AS f ON u."user_id" = f."user_id" Left JOIN "social"
AS so ON u."user_id" = so."user_id" WHERE u.user_id=1;
```

Esta consulta retorna dados sociais de um único usuário agrupados em uma única tabela.

Dados médicos

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, h.height, h.weight, h.allergies, h.medical_insurance,
h.medical_insurance_register, h.hospital_for_removal, h.private_physician_name,
h.private_physician_phone, h.private_physician_email, m.medicine_name,
m.medicine_dosage, mh.medical_history_resume, me.ecg, me.pa, me.sat_o2,
me.glycemia, me.other_exams FROM "user_register" AS u Left JOIN "city"
AS ci ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
Left JOIN "health" AS h ON u."user_id" = h."user_id" Left JOIN "medicine"
AS m ON u."user_id" = m."user_id" Left JOIN "medical_history" AS mh
ON u."user_id" = mh."user_id" Left JOIN "medical_exams" AS me ON
u."user_id" = me."user_id";
```

Esta consulta retorna dados médicos dos usuários agrupados em uma única tabela.

Dados médicos por usuário

```
SELECT u.user_id, u.name AS user_name, u.last_name AS user_last_name
,u.email AS user_email, u.cpf AS user_cpf, u.rg AS user_rg ,u.password AS
user_pass, u.birth_date AS user_birth_date, u.address AS user_address,
u.number AS user_number, u.address_complement AS user_address_complement,
u.postal_code AS user_postal_code, u.phone AS user_phone, u.mobile_phone
AS user_mobile_phone, ci.name AS city_name, st.name AS state_name,
co.name AS country_name, h.height, h.weight, h.allergies, h.medical_insurance,
h.medical_insurance_register, h.hospital_for_removal, h.private_physician_name,
h.private_physician_phone, h.private_physician_email, m.medicine_name,
m.medicine_dosage, mh.medical_history_resume, me.ecg, me.pa, me.sat_o2,
me.glycemia, me.other_exams FROM "user_register" AS u Left JOIN "city"
```

```
AS ci ON u."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
Left JOIN "health" AS h ON u."user_id" = h."user_id" Left JOIN "medicine"
AS m ON u."user_id" = m."user_id" Left JOIN "medical_history" AS mh
ON u."user_id" = mh."user_id" Left JOIN "medical_exams" AS me ON
u."user_id" = me."user_id" WHERE u.user_id=1;
```

Esta consulta retorna dados médicos de um único usuário agrupados em uma única tabela.

Informações de empresa, condomínio e propriedades

Dados de empresa

```
SELECT cm.company_id, cm.name, cm.cnpj, cm.address, cm.number,
cm.city_id, cm.address_complement, cm.phone, cm.email, cm.website,
cm.company_resume, ci.name AS city_name, st.name AS state_name,
co.name AS country_name FROM "company" AS cm Left JOIN "city"
AS ci ON cm."city_id" = ci."city_id" Left JOIN "state" AS st ON
ci."state_id" = st."state_id" Left JOIN "country" AS co ON st."country_id"
= co."country_id";
```

Esta consulta retorna dados de empresas agrupadas em uma única tabela.

Dados de empresa por empresa

```
SELECT cm.company_id, cm.name, cm.cnpj, cm.address, cm.number,
cm.city_id, cm.address_complement, cm.phone, cm.email, cm.website,
cm.company_resume, ci.name AS city_name, st.name AS state_name,
co.name AS country_name FROM "company" AS cm Left JOIN "city" AS ci
ON cm."city_id" = ci."city_id" Left JOIN "state" AS st ON ci."state_id" =
st."state_id" Left JOIN "country" AS co ON st."country_id" = co."country_id"
WHERE cm.company_id=1;
```

Esta consulta retorna dados de uma única empresa agrupados em uma única tabela.

Dados de propriedades e condomínio

```
SELECT pt.property_type, ct.condominium_type, pr.address, pr.number,
pr.address_complement, pr.area, pr.rooms, pr.bathrooms, pr.parking_spaces,
pr.floor, ci.name AS city_name, st.name AS state_name, co.name AS coun-
try_name, cd.name AS condominium_name, cd.cnpj AS condominium_cnpj,
cd.address AS condominium_address, cd.number AS condominium_number,
cd.address_complement AS condominium_address_complement, cd.phone
AS condominium_phone, cd.email AS condominium_email, cd.website AS
```

condominium_website

```
FROM "property" AS pr Left JOIN "property_type" AS pt ON pr."property_type"
= pt."property_type_id" Left JOIN "city" AS ci ON pr."city_id" =
ci."city_id" Left JOIN "condominium" AS cd ON pr."condominium_id"
= cd."condominium_id" Left JOIN "condominium_type" AS ct ON
cd."condominium_type" = ct."condominium_type_id" Left JOIN "state"
AS st ON ci."state_id" = st."state_id" Left JOIN "country" AS co ON
st."country_id" = co."country_id";
```

Esta consulta retorna dados de propriedades agrupados em uma única tabela.

Dados de propriedades e condomínio por propriedade

```
SELECT pt.property_type, ct.condominium_type, pr.address, pr.number,
pr.address_complement, pr.area, pr.rooms, pr.bathrooms, pr.parking_spaces,
pr.floor, ci.name AS city_name, st.name AS state_name, co.name AS coun-
try_name, cd.name AS condominium_name, cd.cnpj AS condominium_cnpj,
cd.address AS condominium_address, cd.number AS condominium_number,
cd.address_complement AS condominium_address_complement, cd.phone
AS condominium_phone, cd.email AS condominium_email, cd.website AS
condominium_website
FROM "property" AS pr Left JOIN "property_type" AS pt ON pr."property_type"
= pt."property_type_id" Left JOIN "city" AS ci ON pr."city_id" =
ci."city_id" Left JOIN "condominium" AS cd ON pr."condominium_id"
= cd."condominium_id" Left JOIN "condominium_type" AS ct ON
cd."condominium_type" = ct."condominium_type_id" Left JOIN "state"
AS st ON ci."state_id" = st."state_id" Left JOIN "country" AS co ON
st."country_id" = co."country_id" WHERE pr.property_id=1;
```

Esta consulta retorna dados de uma única propriedade agrupados em uma única tabela.