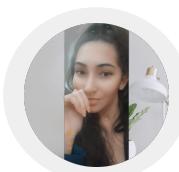
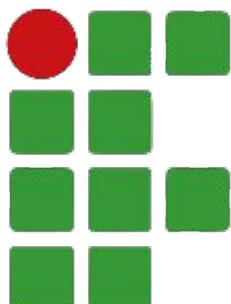


BANCO DE DADOS 2



Projeto de Banco Financeiro em PL SQL.



Emily



Leonardo



Vinicius



OBJETIVO DO PROJETO

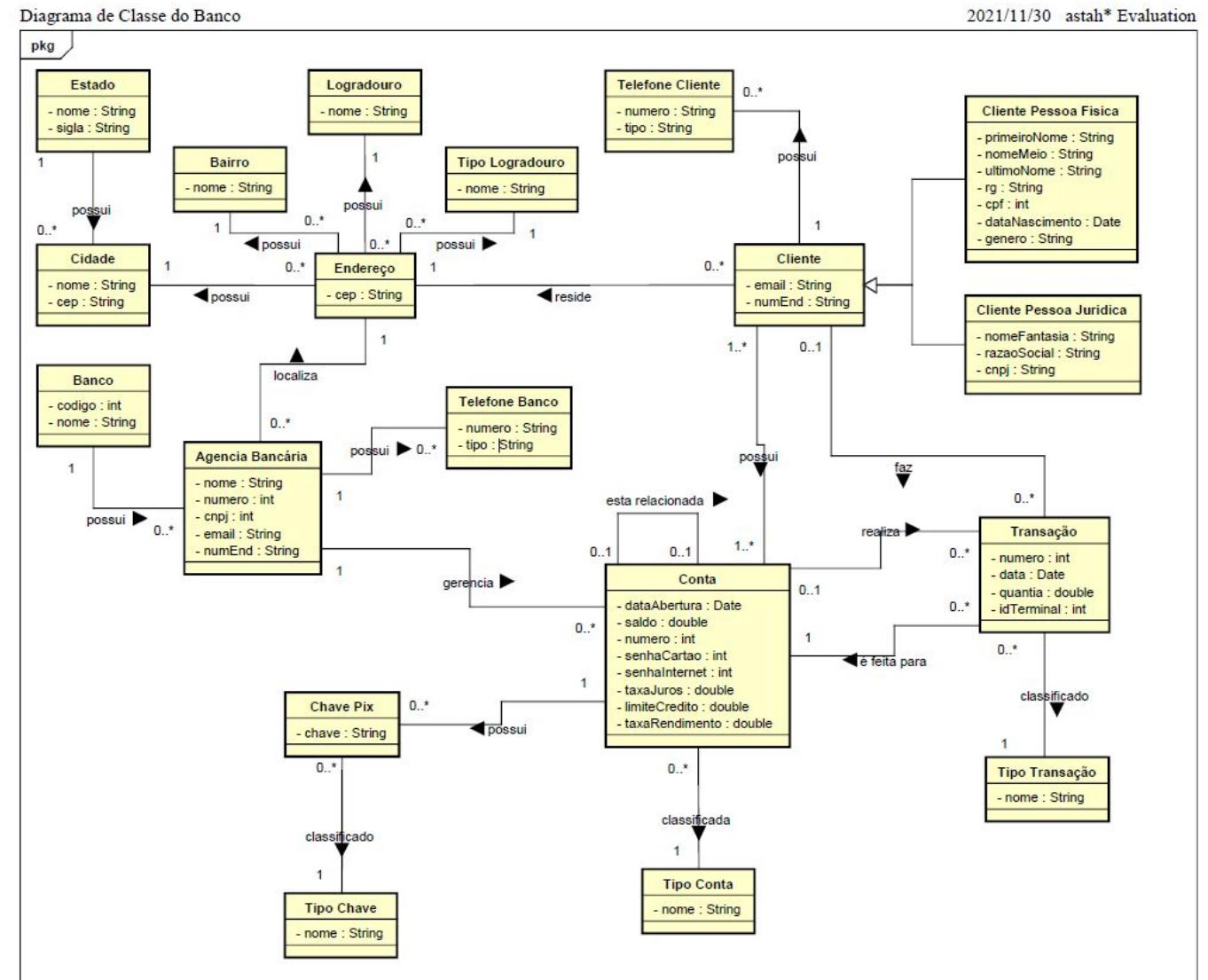


Nosso projeto tem como objetivo aplicar os conhecimentos adquiridos durante o período de um semestre na disciplina de banco de dados 2.

O projeto retrata um banco de dados utilizado para registrar e administrar os registros bancários de uma determinada região, onde é possível visualizar de uma forma geral cada cadastro, seja de clientes, transações, endereço, banco, e seus detalhes, como a conta de um cliente, o seu tipo, suas chaves pix, sua agência, entre outros detalhes.

Diagrama de Classe

Nesse tópico vamos apresentar como seria por dentro do projeto do banco, aqui vemos a comunicação interna entre todas as tabelas da base de dados do banco aqui vemos as transições de dados entre uma classe e outra tornando junto um conjunto de dados que dá início ao nosso projeto de banco de dados com ênfase em banco financeiro



Estrutura das Tabelas - Estado

Aqui vemos a estrutura da Tabela estado onde temos 3 campos de verificação para not null, e duas chaves unicas

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(100)	Sim				1	1
sigla	varchar2(2)	Sim				2	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	índice	Inválido
ESTADO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	25/11/21 19:17:29	-	-
ESTADO_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	25/11/21 19:17:29	-	-
ESTADO_CK_03	Verificação	SIGLA IS NOT NULL	-	-	-	ENABLED	25/11/21 19:17:29	-	-
ESTADO_PK	Primária	-	-	ID	-	ENABLED	25/11/21 19:17:29	ESTADO_PK	-
ESTADO_UK_01	Único	-	-	NOME	-	ENABLED	25/11/21 19:17:29	ESTADO_UK_01	-
ESTADO_UK_02	Único	-	-	SIGLA	-	ENABLED	25/11/21 19:17:29	ESTADO_UK_02	-

Estrutura das Tabelas - Cidade

Aqui vemos a estrutura da Tabela Cidade, onde é cadastrado as cidades com a chave estrangeira vinda da tabela estado , nela possui 3 campos de verificação e 1 campo único

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(200)	Sim				1	1
cep	varchar2(11)	Não				2	1
idEstado	number	Sim		Estado	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
CIDADE_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	25/11/21 19:22:17	-
CIDADE_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	25/11/21 19:22:17	-
CIDADE_CK_03	Verificação	ID_ESTADO IS NOT NULL	-	-	-	ENABLED	25/11/21 19:22:17	-
CIDADE_FK_IDESTADO	Estrangeira	-	CIDADE_FK_IDESTADO (WKSP_BANCOE.V_ESTADO)	ID_ESTADO	NO ACTION	ENABLED	25/11/21 19:22:17	-
CIDADE_PK	Primária	-	-	ID	-	ENABLED	25/11/21 19:22:17	CIDADE_PK
CIDADE_UK_01	Único	-	-	NOME, ID_ESTADO	-	ENABLED	25/11/21 19:22:17	CIDADE_UK_01

Estrutura das Tabelas - Bairro

Nessa tabela podemos observar que só a dois campos um pro identificador da tabela e outro para inserir o nome as próximas duas tabelas a seguir seguirá esse mesmo padrão com apenas a chave única no nome e dois verificadores de campos nulos

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(200)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
BAIRRO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 03:19:03	-
BAIRRO_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 03:19:13	-
BAIRRO_PK	Primária	-	-	ID	-	ENABLED	01/12/21 00:30:05	BAIRRO_PK
BAIRRO_UK_01	Único	-	-	NOME	-	DISABLED	11/12/21 04:05:55	-

Estrutura das Tabelas

- Logradouro

Como mencionado anteriormente nessa tabela de logradouro segue com a mesma estrutura contendo apenas o identificador e seu campo para nome seus dois campo de verificação de campos nulos e seu campo de único.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(200)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	índice
LOGRADOURO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	01/12/21 00:32:09	-
LOGRADOURO_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	01/12/21 00:32:09	-
LOGRADOURO_PK	Primária	-	-	ID	-	ENABLED	01/12/21 00:32:09	LOGRADOURO_PK
LOGRADOURO_UK_01	Único	-	-	NOME	-	ENABLED	01/12/21 18:14:14	LOGRADOURO_UK_01

Estrutura das Tabelas - Tipo logradouro

Nessa tabela permanece a estrutura das suas anteriores contendo seu Identificador seu campo de nome e seus verificadores e campos únicos.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(200)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TIPOLOGRADOURO_PK	Primária	-	-	ID	-	ENABLED	01/12/21 00:34:28	TIPOLOGRADOURO_PK
TIPO_LOGRADOURO_CK_01	Verificação	"ID" IS NOT NULL	-	-	-	ENABLED	08/12/21 02:45:46	-
TIPO_LOGRADOURO_CK_02	Verificação	"NOME" IS NOT NULL	-	-	-	ENABLED	08/12/21 02:46:50	-
TIPO_LOGRADOURO_UK_01	Único	-	-	NOME	-	ENABLED	08/12/21 02:50:06	TIPO_LOGRADOURO_UK_01

Estrutura das Tabelas - Endereço

Nessa tabela endereço e onde fica armazenado todo o endereço dos clientes e bancos nelas recebemos 4 chaves estrangeiras vinda de diversas tabelas como cidade, bairro, logradouro etc também temos seus 5 campos de verificador de campo nulo e sua única chave única no campo CEP.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
cep	varchar2(11)	Não				1	1
idCidade	number	Sim		Cidade	id		
idBairro	number	Sim		Bairro	id		
idLogradouro	number	Sim		Logradouro	id		
idTipoLogradouro	number	Sim		Tipo Logradouro	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
ENDERECO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 02:25:17	-
ENDERECO_CK_02	Verificação	ID_CIDADE IS NOT NULL	-	-	-	ENABLED	08/12/21 03:17:09	-
ENDERECO_CK_03	Verificação	ID_BAIRRO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:17:32	-
ENDERECO_CK_04	Verificação	ID_LOGRADOURO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:17:44	-
ENDERECO_CK_05	Verificação	ID_TIPO_LOGRADOURO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:18:01	-
ENDERECO_FK_IDBAIRRO	Estrangeira	-	ENDERECO_FK_IDBAIRRO (WKSP_BANCOEV.BAIRRO)	ID_BAIRRO	NO ACTION	ENABLED	01/12/21 00:44:16	-
ENDERECO_FK_IDCIDADE	Estrangeira	-	ENDERECO_FK_IDCIDADE (WKSP_BANCOEV.CIDADE)	ID_CIDADE	NO ACTION	ENABLED	01/12/21 00:44:16	-
ENDERECO_FK_IDLOGRADOURO	Estrangeira	-	ENDERECO_FK_IDLOGRADOURO (WKSP_BANCOEV.LOGRADOURO)	ID_LOGRADOURO	NO ACTION	ENABLED	01/12/21 00:44:16	-
ENDERECO_FK_TIPOLOGRADOURO	Estrangeira	-	ENDERECO_FK_TIPOLOGRADOURO (WKSP_BANCOEV.TIPO_LOGRADOURO)	ID_TIPO_LOGRADOURO	NO ACTION	ENABLED	01/12/21 00:44:16	-
ENDERECO_PK	Primária	-	-	ID	-	ENABLED	01/12/21 00:44:16	ENDERECO_PK
ENDERECO_UK_01	Único	-	-	CEP	-	ENABLED	08/12/21 02:26:53	ENDERECO_UK_01

Estrutura das Tabelas - Banco

Já nessa tabela e cadastrada às informações do banco como seu código único e seu nome contendo duas chaves únicas no nome e no código e duas chaves de verificação de campos nulos

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
código	number	Sim				1	1
nome	varchar2(100)	Sim				2	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
BANCO_CK_01	Verificação	CODIGO IS NOT NULL	-	-	-	ENABLED	01/12/21 00:50:41	-
BANCO_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	01/12/21 00:50:46	-
BANCO_PK	Primária	-	-	ID	-	ENABLED	01/12/21 00:50:05	BANCO_PK
BANCO_UK_01	Único	-	-	CODIGO, NOME	-	ENABLED	01/12/21 18:20:05	BANCO_UK_01

Estrutura das Tabelas - Agência Bancária

Nesta tabela temos armazenadas as informações sobre as agências bancárias nela identificamos o banco com a chave estrangeira e o endereço também com a chave estrangeira nela também temos várias chaves únicas como pode ver na modelagem junto com seus 8 campo de verificação de campo nulo.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
número	number	Sim				3	2
nome	varchar2(100)	Sim				1	2
cnpj	varchar(18)	Sim				2	1
email	varchar2(100)	Sim					
numEnd	varchar2(10)	Sim					
idBanco	number	Sim		Banco	id	1/3	1
idEndereco	number	Sim		Endereco	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
AGENCIA_BANCARIA_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_02	Verificação	NUMERO IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_03	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_04	Verificação	CNPJ IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_05	Verificação	EMAIL IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_06	Verificação	NUM_END IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_07	Verificação	ID_BANCO IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_CK_08	Verificação	ID_ENDERECO IS NOT NULL	-	-	-	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_FK_BANCO	Estrangeira	-	AGENCIA_BANCARIA_FK_BANCO (WKSP_BANCOEV.BANCO)	ID_BANCO	NO ACTION	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_FK_ENDERECO	Estrangeira	-	AGENCIA_BANCARIA_FK_ENDERECO (WKSP_BANCOEV.ENDEREKO)	ID_ENDERECO	NO ACTION	ENABLED	08/12/21 01:55:50	-
AGENCIA_BANCARIA_PK	Primária	-	-	ID	-	ENABLED	08/12/21 01:55:50	AGENCIA_BANCARIA_PK
AGENCIA_BANCARIA_UK_01	Único	-	-	ID_BANCO, NOME	-	ENABLED	08/12/21 01:55:50	AGENCIA_BANCARIA_UK_01
AGENCIA_BANCARIA_UK_02	Único	-	-	CNPJ	-	ENABLED	08/12/21 01:55:50	AGENCIA_BANCARIA_UK_02
AGENCIA_BANCARIA_UK_03	Único	-	-	ID_BANCO, NUMERO	-	ENABLED	08/12/21 01:55:50	AGENCIA_BANCARIA_UK_03

Estrutura das Tabelas - Telefone Banco

Nessa tabela fica a estrutura onde armazenamos os telefones referente às agências bancárias por isso ela pega uma chave estrangeira de la e possuí seus 4 campos com o verificador de campos nulos.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
número	varchar2(13)	Sim				1	1
tipo	varchar2(20)	Sim					
idAgencia Bancaria	number	Sim		Agência Bancária	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TELEFONE_BANCO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 03:00:26	-
TELEFONE_BANCO_CK_02	Verificação	NUMERO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:00:26	-
TELEFONE_BANCO_CK_03	Verificação	TIPO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:00:26	-
TELEFONE_BANCO_CK_04	Verificação	ID_AGENCIA_BANCARIA IS NOT NULL	-	-	-	ENABLED	08/12/21 03:00:26	-
TELEFONE_BANCO_FK_AGENCIA_BANCARIA	Estrangeira	-	TELEFONE_BANCO_FK_AGENCIA_BANCARIA (WKBSP_BANCOEVAGENCIABANCARIA)	ID_AGENCIA_BANCARIA	NO ACTION	ENABLED	08/12/21 03:00:26	-
TELEFONE_BANCO_PK	Primária	-	-	ID	-	ENABLED	08/12/21 03:00:26	TELEFONE_BANCO_PK
TELEFONE_BANCO_UK_01	Único	-	-	NUMERO	-	ENABLED	08/12/21 03:00:26	TELEFONE_BANCO_UK_01

Estrutura das Tabelas - Cliente

Nessa tabela Cliente recebemos além do identificador dela um campo onde pegamos o endereço do cliente tanto físico quanto jurídico o numeral da localidade e o email e nela iremos trabalhar com várias tabelas que precisa relacionar a um cliente.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
email	varchar2(100)	Não					
numEnd	varchar2(10)	Sim					
idEndereco	number	Sim		Endereço	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
CLIENTE_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	06/12/21 17:01:47	-
CLIENTE_CK_02	Verificação	NUM_END IS NOT NULL	-	-	-	ENABLED	08/12/21 03:06:47	-
CLIENTE_CK_03	Verificação	ID_ENDERECO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:07:44	-
CLIENTE_FK	Estrangeira	-	CLIENTE_FK (WKSP_BANCOEV.ENDERECO)	ID_ENDERECO	NO ACTION	ENABLED	06/12/21 17:01:47	-
CLIENTE_PK	Primária	-	-	ID	-	ENABLED	06/12/21 17:01:47	CLIENTE_PK

Estrutura das Tabelas - Cliente Pessoa Física

Nesta Estrutura da tabela armazenamos as informações de pessoas físicas por isso nela usamos como chave estrangeira a tabela cliente onde já armazenamos informações como email e endereço por isso nessa tabela os campos são só relacionados a pessoa física como nome, documentos e genero..

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
primeiroNome	varchar2(50)	Sim					
segundoNome	varchar2(50)	Sim					
ultimoNome	varchar2(50)	Sim					
rg	varchar2(12)	Sim				1	1
cpf	number	Sim				2	1
dataNascimento	date	Sim					
genero	varchar2(20)	Sim					
idcliente	number	Sim		Cliente	id	3	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
CLIENTE_PESSOA_FISICA_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 03:04:53	-
CLIENTE_PESSOA_FISICA_CK_02	Verificação	PRIMEIRO_NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 03:05:16	-
CLIENTE_PESSOA_FISICA_CK_03	Verificação	SEGUNDO_NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 03:05:55	-
CLIENTE_PESSOA_FISICA_CK_04	Verificação	ULTIMO_NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 03:06:12	-
CLIENTE_PESSOA_FISICA_CK_05	Verificação	RG IS NOT NULL	-	-	-	ENABLED	08/12/21 03:06:34	-
CLIENTE_PESSOA_FISICA_CK_06	Verificação	CPF IS NOT NULL	-	-	-	ENABLED	08/12/21 03:06:59	-
CLIENTE_PESSOA_FISICA_CK_07	Verificação	DATA_NASCIMENTO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:07:15	-
CLIENTE_PESSOA_FISICA_CK_08	Verificação	GENERO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:07:27	-
CLIENTE_PESSOA_FISICA_CK_09	Verificação	ID_CLIENTE IS NOT NULL	-	-	-	ENABLED	08/12/21 03:07:41	-
CLIENTE_PESSOA_FISICA_FK	Estrangeira	-	CLIENTE_PESSOA_FISICA_FK (WKSP_BANCOEV.CLIENTE)	ID_CLIENTE	NO ACTION	ENABLED	08/12/21 03:12:56	-
CLIENTE_PESSOA_FISICA_PK	Primária	-	-	ID	-	ENABLED	08/12/21 03:13:39	CLIENTE_PESSOA_FISICA_PK
CLIENTE_PESSOA_FISICA_UK_01	Único	-	-	RG	-	ENABLED	08/12/21 03:09:39	CLIENTE_PESSOA_FISICA_UK_01
CLIENTE_PESSOA_FISICA_UK_02	Único	-	-	CPF	-	ENABLED	08/12/21 03:10:07	CLIENTE_PESSOA_FISICA_UK_02
CLIENTE_PESSOA_FISICA_UK_03	Único	-	-	ID_CLIENTE	-	ENABLED	08/12/21 03:11:55	CLIENTE_PESSOA_FISICA_UK_03

Estrutura das Tabelas - Cliente Pessoa Jurídica

Com o mesmo propósito da tabela anterior porém diferentes na tabela pessoa jurídica também pegamos de fora informações como email e endereço e nela armazenamos apenas informações pertence a uma entidade jurídica como CNPJ, Razão e entre outros.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nomeFantasia	varchar2(100)	Sim					
razaoSocial	varchar2(100)	Sim					
cnpj	varchar2(17)	Sim				1	1
idCliente	number	Sim		Cliente	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
CLIENTE_PESSOA_JURIDICA_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 03:23:15	-
CLIENTE_PESSOA_JURIDICA_CK_02	Verificação	NOME_FANTASIA IS NOT NULL	-	-	-	ENABLED	08/12/21 03:24:40	-
CLIENTE_PESSOA_JURIDICA_CK_03	Verificação	RAZAO_SOCIAL IS NOT NULL	-	-	-	ENABLED	08/12/21 03:24:53	-
CLIENTE_PESSOA_JURIDICA_CK_04	Verificação	CNPJ IS NOT NULL	-	-	-	ENABLED	08/12/21 03:25:05	-
CLIENTE_PESSOA_JURIDICA_CK_05	Verificação	ID_CLIENTE IS NOT NULL	-	-	-	ENABLED	08/12/21 03:25:15	-
CLIENTE_PESSOA_JURIDICA_FK	Estrangeira	-	CLIENTE_PESSOA_JURIDICA_FK (WKSP_BANCOEV.CLIENTE)	ID_CLIENTE	NO ACTION	ENABLED	08/12/21 03:26:28	-
CLIENTE_PESSOA_JURIDICA_PK	Primária	-	-	ID	-	ENABLED	08/12/21 03:26:46	CLIENTE_PESSOA_JURIDICA_PK

Estrutura das Tabelas - Telefone Cliente

Nesta tabela vemos semelhança com a estrutura do telefone banco porém essa tabela é especialmente feita para armazenar telefone de clientes independente do seu tipo, por isso nela temos 3 campos com verificação de campo nulo e nela também temos uma chave estrangeira vinda lá da tabela cliente.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
número	varchar2(13)	Sim				1	1
tipo	varchar2(20)	Sim					
idCliente	number	Sim		Cliente	id	1	2

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TELEFONE_CLIENTE_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 03:56:28	-
TELEFONE_CLIENTE_CK_02	Verificação	NUMERO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:56:50	-
TELEFONE_CLIENTE_CK_03	Verificação	TIPO IS NOT NULL	-	-	-	ENABLED	08/12/21 03:57:04	-
TELEFONE_CLIENTE_FK	Estrangeira	-	TELEFONE_CLIENTE_FK (WKSP_BANCOEV.CLIENTE)	ID_CLIENTE	NO ACTION	ENABLED	08/12/21 03:57:49	-
TELEFONE_CLIENTE_PK	Primária	-	-	ID	-	ENABLED	08/12/21 03:34:50	TELEFONE_CLIENTE_PK

Estrutura das Tabelas - Conta

Essa tabela é uma das mais importante do projeto é nela que armazenamos as informações mais importante e é nela que vamos usar para fazer as transações de conta e entre uma conta e outra por isso nela temos vários campos de verificação de campo nulo e dois campos unicos no numero da conta e agencia.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
dataAbertura	date	Sim					
numero	number	Sim				1	2
saldo	float	Sim					
senhaCartao	number	Sim					
senhalInternet	number	Sim					
taxajuros	float	Não					
limiteCredito	float	Não					
taxaRendimento	float	Não					
idAgenciaBancaria	number	Sim		Agência Bancária	id	1	1
idCliente	number	Sim		Cliente	id		
idConta	number	Não		Conta	id		
idTipoconta	number	Sim		Tipo Conta	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
CONTA_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_02	Verificação	DATA_ABERTURA IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_03	Verificação	NUMERO IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_04	Verificação	SALDO IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_05	Verificação	SENHA_CARTAO IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_06	Verificação	SENHA_INTERNET IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_07	Verificação	ID_AGENCIA_BANCARIA IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_08	Verificação	ID_CLIENTE IS NOT NULL	-	-	-	ENABLED	09/12/21 19:59:07	-
CONTA_CK_09	Verificação	ID_TIPO_CONTA IS NOT NULL	-	-	-	ENABLED	09/12/21 20:03:34	-
CONTA_FK_AGENCIA_BANCARIA	Estrangeira	-	CONTA_FK_AGENCIA_BANCARIA (WKSP_BANCOEV.AGENCIA_BANCARIA)	ID_AGENCIA_BANCARIA	NO ACTION	ENABLED	09/12/21 19:59:07	-
CONTA_FK_CLIENTE	Estrangeira	-	CONTA_FK_CLIENTE (WKSP_BANCOEV.CLIENTE)	ID_CLIENTE	NO ACTION	ENABLED	09/12/21 19:59:07	-
CONTA_FK_CONTA	Estrangeira	-	CONTA_FK_CONTA (WKSP_BANCOEV.CONTA)	ID_CONTA	NO ACTION	ENABLED	09/12/21 20:00:20	-
CONTA_FK_TIPO_CONTA	Estrangeira	-	CONTA_FK_TIPO_CONTA (WKSP_BANCOEV.TIPO_CONTA)	ID_TIPO_CONTA	NO ACTION	ENABLED	09/12/21 19:59:07	-
CONTA_PK	Primária	-	-	ID	-	ENABLED	09/12/21 19:59:07	CONTA_PK
CONTA_UK_01	Único	-	-	ID_AGENCIA_BANCARIA, NUMERO	-	ENABLED	09/12/21 19:59:08	CONTA_UK_01

Estrutura das Tabelas - Tipo de Conta

Como podemos perceber nessa estrutura simples nós armazenamos as informações dos tipo de conta que cada banco pode trabalhar, nesse projeto vamos utilizar só duas a Corrente e Poupança porém pode ser usadas conta salario, conta de investimento e entre outras, por isso nela só temos um campo único para o nome e dois campos verificadores de campos nulos.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(20)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TIPO_CONTA_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	08/12/21 01:21:00	-
TIPO_CONTA_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	08/12/21 01:21:00	-
TIPO_CONTA_PK	Primária	-	-	ID	-	ENABLED	08/12/21 01:21:00	TIPO_CONTA_PK
TIPO_CONTA_UK_01	Único	-	-	NOME	-	ENABLED	08/12/21 01:21:00	TIPO_CONTA_UK_01

Estrutura das Tabelas - Chave Pix

Nesta tabela fizemos um estrutura que guarde informações de chave pix que hoje é utilizado muito nas instituições financeiras como uma forma de realizar transação e armazenados informações como o identificador da conta que é uma chave estrangeira e fora o identificador do tipo de chave pix que é um campo único.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
chave	varchar2(100)	Sim				1	1
idConta	number	Sim		Conta	id		
idTipoChave	number	Sim		Tipo Chave	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	índice
CHAVE_PIX_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_CK_02	Verificação	CHAVE IS NOT NULL	-	-	-	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_CK_03	Verificação	ID_CONTA IS NOT NULL	-	-	-	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_CK_04	Verificação	ID_TIPO_CHAVE IS NOT NULL	-	-	-	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_FK_CONTA	Estrangeira	-	CHAVE_PIX_FK_CONTA (WKSP_BANCOEV.CONTA)	ID_CONTA	NO ACTION	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_FK_TIPO_CHAVE	Estrangeira	-	CHAVE_PIX_FK_TIPO_CHAVE (WKSP_BANCOEV.TIPO_CHAVE)	ID_TIPO_CHAVE	NO ACTION	ENABLED	09/12/2120:11:58	-
CHAVE_PIX_PK	Primária	-	-	ID	-	ENABLED	09/12/2120:11:58	CHAVE_PIX_PK
CHAVE_PIX_UK_01	Único	-	-	CHAVE	-	ENABLED	09/12/2120:11:58	CHAVE_PIX_UK_01

Estrutura das Tabelas - Tipo de Chave PIX

Nesta tabela armazenamos informações complementares da chave pix com o seu tipo pode ser um CPF, E-Mail ou até mesmo uma chave aleatória por isso em sua estrutura usamos 2 campos de verificação de campos nulos e um campo único.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(100)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TIPO_CHAVE_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	09/12/21 19:44:10	-
TIPO_CHAVE_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	09/12/21 19:44:10	-
TIPO_CHAVE_PK	Primária	-	-	ID	-	ENABLED	09/12/21 19:44:10	TIPO_CHAVE_PK
TIPO_CHAVE_UK_01	Único	-	-	NOME	-	ENABLED	09/12/21 19:44:10	TIPO_CHAVE_UK_01

Estrutura das Tabelas - Transação

Nessa tabela e armazenadas as transações de todos os tipo como deposito, saques e transferências nela usamos duas estrangeiras para referenciar as contas de origem e destino das transacao caso precise e nela também usamos o ID TERMINAL para dizer pro cliente no extrato dele em qual terminal ele realizou essa transação.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
número	number	Sim				1	1
data	date	Sim					
valor	double	Sim					
idTerminal	number	Não					
idTipoTransação	number	Sim		Tipo Transação	id		
idCliente	number	Não		Cliente	id		
<u>idContaOrigem</u>	number	Não		Conta	id		
idContaDestino	number	Sim		Conta	id		

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TRANSACAO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	09/12/21 20:28:13	-
TRANSACAO_CK_02	Verificação	NUMERO IS NOT NULL	-	-	-	ENABLED	09/12/21 20:28:13	-
TRANSACAO_CK_03	Verificação	DATA IS NOT NULL	-	-	-	ENABLED	09/12/21 20:28:13	-
TRANSACAO_CK_04	Verificação	VALOR IS NOT NULL	-	-	-	ENABLED	09/12/21 20:28:13	-
TRANSACAO_CK_05	Verificação	ID_TIPO_TRANSACAO IS NOT NULL	-	-	-	ENABLED	09/12/21 20:28:13	-
TRANSACAO_CK_06	Verificação	ID_CONTA_DESTINO IS NOT NULL	-	-	-	DISABLED	12/12/21 16:03:32	-
TRANSACAO_FK_CLIENTE	Estrangeira	-	TRANSACAO_FK_CLIENTE (WKSP_BANCOEV.CLIENTE)	ID_CLIENTE	NO ACTION	ENABLED	09/12/21 20:28:13	-
TRANSACAO_FK_CONTA_DESTINO	Estrangeira	-	TRANSACAO_FK_CONTA_DESTINO (WKSP_BANCOEV.CONTA)	ID_CONTA_DESTINO	NO ACTION	ENABLED	09/12/21 20:28:13	-
TRANSACAO_FK_CONTA_ORIGEM	Estrangeira	-	TRANSACAO_FK_CONTA_ORIGEM (WKSP_BANCOEV.CONTA)	ID_CONTA_ORIGEM	NO ACTION	ENABLED	09/12/21 20:28:13	-
TRANSACAO_FK_TIPO_TRANSACAO	Estrangeira	-	TRANSACAO_FK_TIPO_TRANSACAO (WKSP_BANCOEV.TIPO_TRANSACAO)	ID_TIPO_TRANSACAO	NO ACTION	ENABLED	09/12/21 20:28:13	-
TRANSACAO_PK	Primária	-	-	ID	-	ENABLED	09/12/21 20:28:12	TRANSACAO_PK
TRANSACAO_UK_01	Único	-	-	NUMERO	-	ENABLED	09/12/21 20:28:13	TRANSACAO_UK_01

Estrutura das Tabelas - Tipo de Transação

Semelhante a tabela anterior nessa tabelas utilizamos a mesma estrutura com 2 campos verificadores de campos nulos e um campo único porém é utilizada para armazenar os tipos de transação como: depósitos, saques e transferências bancárias.

Campo	Tipo	Obrigatório?	Chave Primária?	Chave Estrangeira		Chave Única	
				Tabela	Campo	Grupo	Ordem
id	number	Sim	Sim				
nome	varchar2(100)	Sim				1	1

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice
TIPO_TRANSACAO_CK_01	Verificação	ID IS NOT NULL	-	-	-	ENABLED	09/12/21 19:45:47	-
TIPO_TRANSACAO_CK_02	Verificação	NOME IS NOT NULL	-	-	-	ENABLED	09/12/21 19:45:47	-
TIPO_TRANSACAO_PK	Primária	-	-	ID	-	ENABLED	09/12/21 19:45:46	TIPO_TRANSACAO_PK
TIPO_TRANSACAO_UK_01	Único	-	-	NOME	-	ENABLED	09/12/21 19:45:47	TIPO_TRANSACAO_UK_01

Função: Validação CPF.

Resultados Explicação Descrever Instrução SQL Salva Histórico

FN_VALIDACPF('46281701852')

CPF VALIDO

1 linhas retornadas em 0,02 segundos Fazer Download

```
42      end if;
43      dc_cpf := to_char(dc1_cpf) || to_char(dc2_cpf);
44      if dc_cpf = digito_dig then
45          return 'CPF VALIDO';
46      else
47          return 'CPF INVALIDO';
48      end if;
49  END FN_VALIDACPF;
50
```

Função: Chave Aleatoria Pix.

Resultados Explicação Descrever Instrução SQL Salva Histórico

FN_CHAVEPIX_ALEATORIA('VINICIUSDESOUZASANTOS')

F30B1EBA481DC3BC0D4D0178E815230A

1 linhas retornadas em 0,02 segundos Fazer Download

```
1  create or replace function "FN_CHAVEPIX_ALEATORIA"
2  (pv_valor in VARCHAR2)
3  return VARCHAR2 is
4
5  pv_hexkey varchar2(32) :=null;
6
7  begin
8  pv_hexkey := rawtohex(DBMS_OBFUSCATION_TOOLKIT.md5(input => UTL_RAW.cast_to_raw(pv_valor)));
9  return nvl(pv_hexkey,'');
10 end;
11
```



Procedimento de Cadastro Cliente PF

```
undo | redo | search | A..  
1  create or replace PROCEDURE PR_CAD_CLIENTE_PF (pv_F_NAME IN varchar2,  
2                                               pv_M_NAME IN varchar2,  
3                                               pv_L_NAME IN varchar2,  
4                                               pv_EMAIL IN varchar2,  
5                                               pv_NUM_END IN varchar2,  
6                                               pn_ID_ENDERECHO IN number,  
7                                               pv_RG IN varchar2,  
8                                               pv_CPF IN NUMBER,  
9                                               pd_DATA_NASCIMENTO IN date,  
10                                              pv_genero IN varchar2) IS  
11  vb_existe      BOOLEAN;  
12  vc_cliente     CLIENTE%ROWTYPE;  
13  vc_cliente_pf  CLIENTE_PESSOA_FISICA%ROWTYPE;  
14  vn_id          NUMBER;  
15  vv_erro         VARCHAR2(100);  
16  
17  begin  
18      vc_cliente_pf.PRIMEIRO_NOME    := pv_F_NAME;  
19      vc_cliente_pf.SEGUNDO_NOME    := pv_M_NAME;  
20      vc_cliente_pf.ULTIMO_NOME    := pv_L_NAME;  
21      vc_cliente.EMAIL           := pv_EMAIL;  
22      vc_cliente.NUM_END        := pv_NUM_END;  
23      vc_cliente.ID_ENDERECHO   := pn_ID_ENDERECHO;  
24      vc_cliente_pf.RG           := pv_RG;  
25      vc_cliente_pf.CPF          := pv_CPF;  
26      vc_cliente_pf.DATA_NASCIMENTO := pd_DATA_NASCIMENTO;  
27      vc_cliente_pf.GENERO       := pv_genero;  
28
```

Procedimento de Cadastro Cliente PF

```
17 begin
18     vc_cliente_pf.PRIMEIRO_NOME    := pv_F_NAME;
19     vc_cliente_pf.SEGUNDO_NOME    := pv_M_NAME;
20     vc_cliente_pf.ULTIMO_NOME    := pv_L_NAME;
21     vc_cliente.EMAIL            := pv_EMAIL;
22     vc_cliente.NUM_END          := pv_NUM_END;
23     vc_cliente.ID_ENDERECHO    := pn_ID_ENDERECHO;
24     vc_cliente_pf.RG             := pv_RG;
25     vc_cliente_pf.CPF            := pv_CPF;
26     vc_cliente_pf.DATA_NASCIMENTO := pd_DATA_NASCIMENTO;
27     vc_cliente_pf.GENERO        := pv_genero;
28
29 begin
30     vb_existe := PKG_CLIENTE_PESSOA_FISICA.FN_VERIFICAREXISTE(pn_conteudo => vc_cliente_pf.CPF);
31     IF vb_existe = FALSE THEN
32         PKG_CLIENTE.PR_INCLUIR(pr_cliente => vc_cliente);
33         vn_id := FN_RETORNA_ID_CLIENTE;
34         vc_cliente_pf.ID_CLIENTE := vn_id;
35         PKG_CLIENTE_PESSOA_FISICA.PR_INCLUIR(pr_CLIENTE_PESSOA_FISICA => vc_cliente_pf, pv_erro => vv_erro);
36     END IF;
37 end;
38 end PR_CAD_CLIENTE_PF;
39
```

Procedimento de Cadastro Cliente PJ

```
1 create or replace PROCEDURE PR_CAD_CLIENTE_PJ (pv_EMAIL IN varchar2,
2                                         pv_NUM_END IN varchar2,
3                                         pn_ID_ENDERECHO IN number,
4                                         pv_NOME_FANTASIA IN VARCHAR2,
5                                         pv_RAZAO_SOCILA IN VARCHAR2,
6                                         pv_CNPJ IN VARCHAR2) IS
7     vb_existe    BOOLEAN;
8     vc_cliente   CLIENTE%ROWTYPE;
9     vc_cliente_pj CLIENTE_PESSOA_JURIDICA%ROWTYPE;
10    vn_id        NUMBER;
11    vv_erro       VARCHAR2(100);
12
```

Procedimento de Cadastro Cliente PJ

```
13 begin
14
15     vc_cliente.EMAIL      := pv_EMAIL;
16     vc_cliente.NUM_END    := pv_NUM_END;
17     vc_cliente.ID_ENDERECHO := pn_ID_ENDERECHO;
18     vc_cliente_pj.NOME_FANTASIA := pv_NOME_FANTASIA;
19     vc_cliente_pj.RAZAO_SOCIAL  := pv_RAZAO_SOCILA;
20     vc_cliente_pj.CNPJ       := pv_CNPJ;
21
22 begin
23     vb_existe := PKG_CLIENTE_PESSOA_JURIDICA.FN_VERIFICAREXISTE(pv_conteudo => vc_cliente_pj.CNPJ);
24     IF vb_existe = FALSE THEN
25         PKG_CLIENTE.PR_INCLUIR(pr_cliente => vc_cliente);
26         vn_id := FN_RETORNA_ID_CLIENTE;
27         vc_cliente_pj.ID_CLIENTE := vn_id;
28         PKG_CLIENTE_PESSOA_JURIDICA.PR_INCLUIR(pr_CLIENTE_PESSOA_JURIDICA => vc_cliente_pj, pv_erro => vv_erro);
29     END IF;
30 END;
31 end PR_CAD_CLIENTE_PJ;
32
```

Procedimento de Cadastro de Conta

```
5 C | Q | A::  
1 create or replace PROCEDURE PR_CAD_CONTA (pd_DATA_ABERTURA IN DATE,  
2                                         pn_SALDO IN CONTA.SALDO%TYPE,  
3                                         pn_NUMERO IN NUMBER,  
4                                         pn_SENHA_CARTAO IN NUMBER,  
5                                         pn_SENHA_INTERNET IN NUMBER,  
6                                         pn_TAXA_JUROS IN CONTA.TAXA_JUROS%TYPE,  
7                                         pn_LIMITE_CREDITO IN CONTA.LIMITE_CREDITO%TYPE,  
8                                         pn_TAXA_RENDIMENTO IN CONTA.TAXA_RENDIMENTO%TYPE,  
9                                         pn_ID_AGENCIA_BANCARIA IN NUMBER,  
10                                        pn_ID_CLIENTE IN NUMBER,  
11                                        pn_ID_CONTA IN NUMBER,  
12                                        pn_ID_TIPO_CONTA IN NUMBER) IS  
13   vc_conta      CONTA%ROWTYPE;  
14   vv_erro       VARCHAR2(100);  
15
```

Procedimento de Cadastro de Conta

```
16  < begin
17      vc_conta.DATA_ABERTURA := pd_DATA_ABERTURA;
18      vc_conta.SALDO := pn_SALDO;
19      vc_conta.NUMERO := pn_NUMERO;
20      vc_conta.SENHA_CARTAO := pn_SENHA_CARTAO;
21      vc_conta.SENHA_INTERNET := pn_SENHA_INTERNET;
22      vc_conta.TAXA_JUROS := pn_TAXA_JUROS;
23      vc_conta.LIMITE_CREDITO := pn_LIMITE_CREDITO;
24      vc_conta.TAXA_RENDIMENTO := pn_TAXA_RENDIMENTO;
25      vc_conta.ID_AGENCIA_BANCARIA := pn_ID_AGENCIA_BANCARIA;
26      vc_conta.ID_CLIENTE := pn_ID_CLIENTE;
27      vc_conta.ID_CONTA := pn_ID_CONTA;
28      vc_conta.ID_TIPO_CONTA := pn_ID_TIPO_CONTA;
29
30  < begin
31      | PKG_CONTA.PR_INCLUIR(pr_conta => vc_conta, pv_erro => vv_erro);
32  END;
33 end PR_CAD_CONTA;
34
```

Procedimento de Depósito

```
1  create or replace PROCEDURE PR_DEPOSITO (pn_NUMERO IN varchar2,  
2                                              pd_DATE IN DATE,  
3                                              pn_VALOR IN number,  
4                                              pn_ID_TERMINAL IN number,  
5                                              pn_ID_CLIENTE IN number,  
6                                              pn_ID_CONTA_ORIGEM IN number,  
7                                              pn_ID_CONTA_DESTINO IN number) IS  
8  
9  vt_transacao    TRANSACAO%ROWTYPE;  
10 vc_conta        CONTA%ROWTYPE;
```

Procedimento de Depósito

```
12 begin
13     vt_transacao.NUMERO := pn_NUMERO;
14     vt_transacao.DATA := pd_DATE;
15     vt_transacao.VALOR := pn_VALOR;
16     vt_transacao.ID_TERMINAL := pn_ID_TERMINAL;
17     vt_transacao.ID_CLIENTE := pn_ID_CLIENTE;
18     vt_transacao.ID_CONTA_ORIGEM := pn_ID_CONTA_ORIGEM;
19     vt_transacao.ID_CONTA_DESTINO := pn_ID_CONTA_DESTINO;
20     vt_transacao.ID_TIPO_TRANSACAO := 2;
21
22     PKG_TRANSACAO.PR_INCLUIR(pr_TRANSACAO => vt_TRANSACAO);
23
24     vc_conta := PKG_CONTA.FN_BUSCAR(pn_id => vt_TRANSACAO.ID_CONTA_ORIGEM);
25     vc_conta.SALDO := vc_conta.SALDO + vt_TRANSACAO.VALOR;
26     PKG_CONTA.PR_ALTERAR(pr_CONTA => vc_conta);
27     --insert into TRANSACAO (NUMERO,DATA,VALOR,ID_TERMINAL,ID_TIPO_TRANSACAO,ID_CLIENTE,ID_CONTA_ORIGEM,ID_CONTA_DESTINO) values (pn_NUMERO,
28     --update CONTA set SALDO = SALDO + pn_Valor where ID = pn_ID_CONTA_ORIGEM;
29 end PR_DEPOSITO;
```

Procedimento de Saque

```
↻ C | Q | A::  
1 create or replace PROCEDURE PR_SAQUE (pn_NUMERO IN varchar2,  
2                                     pd_DATE IN DATE,  
3                                     pn_VALOR IN number,  
4                                     pn_ID_TERMINAL IN number,  
5                                     pn_ID_CLIENTE IN number,  
6                                     pn_ID_CONTA_ORIGEM IN number,  
7                                     pn_ID_CONTA_DESTINO IN number) IS  
8  
9     vt_transacao      TRANSACAO%ROWTYPE;  
10    vc_conta          CONTA%ROWTYPE;  
11
```

Procedimento de Saque

```
12 begin
13     vt_transacao.NUMERO := pn_NUMERO;
14     vt_transacao.DATA := pd_DATE;
15     vt_transacao.VALOR := pn_VALOR;
16     vt_transacao.ID_TERMINAL := pn_ID_TERMINAL;
17     vt_transacao.ID_CLIENTE := pn_ID_CLIENTE;
18     vt_transacao.ID_CONTA_ORIGEM := pn_ID_CONTA_ORIGEM;
19     vt_transacao.ID_CONTA_DESTINO := pn_ID_CONTA_DESTINO;
20     vt_transacao.ID_TIPO_TRANSACAO := 1;
21
22     PKG_TRANSACAO.PR_INCLUIR(pr_TRANSACAO => vt_TRANSACAO);
23     vc_conta := PKG_CONTA.FN_BUSCAR(pn_id => vt_TRANSACAO.ID_CONTA_DESTINO);
24     vc_conta.SALDO := vc_conta.SALDO - vt_TRANSACAO.VALOR;
25     PKG_CONTA.PR_ALTERAR(pr_CONTA => vc_conta);
26     --insert into TRANSACAO (NUMERO,DATA,VALOR,ID_TERMINAL,ID_TIPO_TRANSACAO,ID_CLIENTE,ID_CONTA_ORIGEM,ID_CONTA_DESTINO)
27     --update CONTA set SALDO = SALDO - pn_Valor where ID = pn_ID_CONTA_DESTINO;
28 end PR_SAQUE;
29
```

Procedimento de Transferencia Bancaria

```
1  create or replace PROCEDURE PR_TRANSFERENCIA ( pn_NUMERO IN varchar2,
2                                               pd_DATE IN DATE,
3                                               pn_VALOR IN number,
4                                               pn_ID_TERMINAL IN number,
5                                               pn_ID_CLIENTE IN number,
6                                               pn_ID_CONTA_ORIGEM IN number,
7                                               pn_ID_CONTA_DESTINO IN number) IS
8
9
10  vt_transacao      TRANSACAO%ROWTYPE;
11  vc_conta_origem   CONTA%ROWTYPE;
12  vc_conta_destino  CONTA%ROWTYPE;
13
14  begin
15    vt_transacao.NUMERO := pn_NUMERO;
16    vt_transacao.DATA := pd_DATE;
17    vt_transacao.VALOR := pn_VALOR;
18    vt_transacao.ID_TERMINAL := pn_ID_TERMINAL;
19    vt_transacao.ID_CLIENTE := pn_ID_CLIENTE;
20    vt_transacao.ID_CONTA_ORIGEM := pn_ID_CONTA_ORIGEM;
21    vt_transacao.ID_CONTA_DESTINO := pn_ID_CONTA_DESTINO;
22    vt_transacao.ID_TIPO_TRANSACAO := 3;
23
24    PKG_TRANSACAO.PR_INCLUIR(pr_TRANSACAO => vt_transacao);
25
26    vc_conta_origem := PKG_CONTA.FN_BUSCAR(pn_id => vt_transacao.ID_CONTA_DESTINO);
27    vc_conta_origem.SALDO := vc_conta_origem.SALDO - vt_transacao.VALOR;
28
29    PKG_CONTA.PR_ALTERAR(pr_CONTA => vc_conta_origem);
```

Procedimento de Transferencia Bancaria

```
28  PKG_CONTA.PR_ALTERAR(pr_CONTA => vc_conta_origem);
29  --UPDATE CONTA SET SALDO = SALDO - pr_TRANSACAO.VALOR where ID = pr_TRANSACAO.ID_CONTA_ORIGEM;
30  vc_conta_destino := PKG_CONTA.FN_BUSCAR(pn_id => vt_transacao.ID_CONTA_DESTINO);
31  vc_conta_destino.SALDO := vc_conta_destino.SALDO + vt_transacao.VALOR;
32
33  PKG_CONTA.PR_ALTERAR(pr_CONTA => vc_conta_destino);
34  --UPDATE CONTA SET SALDO = SALDO + pr_TRANSACAO.VALOR where ID = pr_TRANSACAO.ID_CONTA_DESTINO;
35
36  --insert into TRANSACAO (NUMERO,DATA,VALOR,ID_TERMINAL,ID_TIPO_TRANSACAO,ID_CLIENTE,ID_CONTA_ORIGEM,ID_CONTA_DESTINO)
37  --update CONTA set SALDO = SALDO - pn_Valor where ID = pn_ID_CONTA_ORIGEM;
38  --update CONTA set SALDO = SALDO + pn_Valor where ID = pn_ID_CONTA_DESTINO;
39  end PR_TRANSFERENCIA;
40
```

Procedimento de Envio de E-mail

```
1  create or replace PROCEDURE PR_ENVIAREMAILAUT ( pv_smtp IN varchar2,pv_usuario IN varchar2,
2                                                 pv_senha IN varchar2,pv_remetente IN varchar2,
3                                                 pv_destinatario IN varchar2, pv_assunto IN varchar2,
4                                                 pc_mensagem IN clob) IS
5  pv_conn utl_smtp.connection;
6
7  begin
8  pv_conn := utl_smtp.open_connection (pv_smtp,587);
9  utl_smtp.ehlo(pv_conn,pv_smtp);
10 utl_smtp.command (pv_conn, 'AUTH LOGIN');
11
12 utl_smtp.command (pv_conn,utl_raw.cast_to_varchar2(utl_encode.base64_encode(utl_raw.cast_to_raw(pv_usuario))));
13 utl_smtp.command (pv_conn,utl_raw.cast_to_varchar2(utl_encode.base64_encode(utl_raw.cast_to_raw(pv_senha))));
14
15 utl_smtp.mail (pv_conn,('<' ||pv_remetente||'>'));
16 utl_smtp.rcpt (pv_conn,('<' ||pv_destinatario||'>'));
17
```

Procedimento de Envio de E-mail

```
17  utl_smtp.open_data(pv_conn);
18  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw('Content-Type: Text/Html; charset=UTF-8'||utl_tcp.crlf));
19  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw('From:'||pv_remetente||utl_tcp.crlf));
20  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw('To:'||pv_destinatario||utl_tcp.crlf));
21  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw('Subject:'||pv_assunto||utl_tcp.crlf));
22  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw(''||utl_tcp.crlf));
23  utl_smtp.write_raw_data(pv_conn, utl_raw.cast_to_raw(utl_tcp.crlf||pc_mensagem));
24
25  utl_smtp.CLOSE_DATA(pv_conn);
26  utl_smtp.QUIT(pv_conn);
27  EXCEPTION
28  WHEN OTHERS THEN
29    utl_smtp.quit (pv_conn);
30    RAISE_APPLICATION_ERROR(-20011,'Não foi Possivel enviar o Email devido ao seguinte erro: '||sqlerrm);
31  end;
32
```

Procedimento de Rendimento para Conta Poupança

```
1  create or replace procedure PR_RENDER_CONTA_POUPANCA
2  is
3  begin
4      UPDATE CONTA SET SALDO = (SALDO * 1.005) WHERE CONTA.ID > 0;
5  end PR_RENDER_CONTA_POUPANCA;
6
```

Pacote de Cliente Pessoa Física

```
5 C | Q | A..  
1  create or replace PACKAGE PKG_CLIENTE_PESSOA_FISICA AS  
2  /* TODO enter package declarations (types, exceptions, methods etc) here */  
3  vr_cliente_Pessoa_Fisica    CLIENTE_PESSOA_FISICA%ROWTYPE;  
4  PROCEDURE PR_INCLUIR        ( pr_cliente_Pessoa_Fisica  IN  CLIENTE_PESSOA_FISICA%ROWTYPE,  
5                                pv_erro          OUT varchar2  
6                                );  
7  PROCEDURE PR_ALTERAR        ( pr_cliente_Pessoa_Fisica  IN  CLIENTE_PESSOA_FISICA%ROWTYPE );  
8  PROCEDURE PR_EXCLUIR        ( pn_id           IN  number );  
9  FUNCTION  FN_BUSCAR         ( pv_conteudo      IN  NUMBER  
10                            ) return CLIENTE_PESSOA_FISICA%ROWTYPE;  
11  FUNCTION  FN_BUSCAR         ( pn_id           IN  NUMBER  
12                            ) return CLIENTE_PESSOA_FISICA%ROWTYPE;  
13  FUNCTION  FN_VERIFICAREXISTE ( pn_conteudo     IN  NUMBER  
14                            ) return boolean;  
15  END PKG_CLIENTE_PESSOA_FISICA;  
16
```

Pacote de Cliente Pessoa Física - Body

```
1  create or replace PACKAGE BODY PKG_CLIENTE_PESSOA_FISICA AS
2      PROCEDURE PR_INCLUIR ( pr_cliente_Pessoa_Fisica IN CLIENTE_PESSOA_FISICA%ROWTYPE,
3                             | pv_erro          OUT varchar2
4                             ) AS
5
6      vb_existe    boolean;
7      vn_id        number;
8
9      BEGIN
10         pv_erro   := '';
11         vb_existe := FN_VERIFICAREXISTE ( pn_conteudo     => pr_cliente_Pessoa_Fisica.CPF);
12         if   vb_existe then
13             pv_erro := 'CLIENTE FÍSICO já cadastrado para o CPF informado';
14         end if;
15
16         if   pv_erro is null then
17             select CLIENTE_PESSOA_FISICA_SEQ.nextval
18                 into  vn_id
19                 from  dual;
20
21             insert into CLIENTE_PESSOA_FISICA (PRIMEIRO_NOME, SEGUNDO_NOME, ULTIMO_NOME, RG, CPF, DATA_NASCIMENTO,
22                                         GENERO, ID_CLIENTE)
23             values          (pr_cliente_Pessoa_Fisica.PRIMEIRO_NOME,
24                               pr_cliente_Pessoa_Fisica.SEGUNDO_NOME,
25                               pr_cliente_Pessoa_Fisica.ULTIMO_NOME,
26                               pr_cliente_Pessoa_Fisica.RG,
27                               pr_cliente_Pessoa_Fisica.CPF,
28                               pr_cliente_Pessoa_Fisica.DATA_NASCIMENTO,
```

Pacote de Cliente Pessoa Física - Body

```
112      vb_existe  boolean;
113      vn_qtde    number;
114
115      cursor Busca_cliente_Pessoa_Fisica_CPF is
116          select count(*)
117          from  CLIENTE_PESSOA_FISICA
118          where  CPF = pn_conteudo;
119
120      BEGIN
121          open  Busca_cliente_Pessoa_Fisica_CPF;
122          fetch Busca_cliente_Pessoa_Fisica_CPF into vn_qtde;
123          if      Busca_cliente_Pessoa_Fisica_CPF%NotFound then
124              vn_qtde := 0;
125          end if;
126          close Busca_cliente_Pessoa_Fisica_CPF;
127
128          if      vn_qtde = 0 then
129              vb_existe := false;
130          else   vb_existe := true;
131          end if;
132
133          return vb_existe;
134      END FN_VERIFICAREXISTE;
135
136  END PKG_CLIENTE_PESSOA_FISICA;
137
```

Job Rendimento da Conta Poupanca

9.1. RENDIMENTO DA CONTA POUPANÇA

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB
(
    JOB_NAME => 'RENDIMENTO_POUPANCA',
    JOB_TYPE => 'STORED_PROCEDURE',
    JOB_ACTION => 'PR_RENDER_CONTA_POUPANCA',
    START_DATE => SYSDATE,
    REPEAT_INTERVAL => 'FREQ=MONTHLY;INTERVAL=1',
    ENABLED => TRUE,
    COMMENTS => 'JOB PARA RENDER AS CONTAS POUPANÇAS '
);
END;
```

Select's do Projeto

Cliente

Extrato 7 dias = (número da conta e agência, nome completo, transações dos últimos 7 dias, saldo)

Extrato 15 dias = (número da conta e agência, nome completo, transações dos últimos 15 dias, saldo)

Extrato 30 dias = (número da conta e agência, nome completo, transações dos últimos 30 dias, saldo)

Saldo = (número da conta e agência, nome completo, saldo)

Agência

Lista de clientes em ordem alfabética

Lista de clientes com conta poupança em ordem alfabética

Lista de clientes com conta corrente em ordem alfabética

Banco

Total de contas por agência

Total de contas poupanças por agência

Total de contas correntes por agência

Total de contas abertas no último mês por agência

Movimentação das agências

Select's (Extrato de 7 Dias e Saldo)

```
SELECT CONTA.NUMERO AS "CONTA",
       AGENCIA_BANCARIA.NUMERO AS "AGÊNCIA BANCÁRIA",
       CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME || ' ' || CLIENTE_PESSOA_FISICA.S
TRANSACAO.NUMERO AS "NUMERO TRANSAÇÃO",
TRANSACAO.DATA,
TRANSACAO.VALOR,
TRANSACAO.ID_TERMINAL AS "TERMINAL",
TIPO_TRANSACAO.NOME AS "TIPO DE TRANSAÇÃO",
TRANSACAO.ID_CONTA_ORIGEM AS "CONTA DE ORIGEM",
TRANSACAO.ID_CONTA_DESTINO AS "CONTA DE DESTINO",
CONTA.SALDO
FROM CONTA,
     AGENCIA_BANCARIA,
     CLIENTE_PESSOA_FISICA,
     TRANSACAO,
     TIPO_TRANSACAO
WHERE CONTA.ID_AGENCIA_BANCARIA = AGENCIA_BANCARIA.ID
      AND CONTA.ID_CLIENTE = CLIENTE_PESSOA_FISICA.ID
      AND TRANSACAO.ID_CLIENTE = CLIENTE_PESSOA_FISICA.ID
      AND TIPO_TRANSACAO.ID = TRANSACAO.ID_TIPO_TRANSACAO
      AND TRANSACAO.DATA > '07-12-2021'
```

```
SELECT CONTA.NUMERO AS "NÚMERO DA CONTA",
       AGENCIA_BANCARIA.NUMERO "NÚMERO DA AGÊNCIA",
       CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME || ' ' || CLIENTE_PESSOA_FISICA.SEGUNDO_NOME || ' ' || CLIENTE_PESSOA_FISICA.TERCEIRO_NOME AS "NOME DO CLIENTE",
       CONTA.SALDO
FROM CONTA,
     AGENCIA_BANCARIA,
     CLIENTE_PESSOA_FISICA
WHERE CONTA.ID_AGENCIA_BANCARIA = AGENCIA_BANCARIA.ID
      AND CONTA.ID_CLIENTE = CLIENTE_PESSOA_FISICA.ID
```

Select's (Lista de Clientes por ordem alfabeticas e por tipo de conta)

```
SELECT CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME || ' ' || CLIENTE_PESSOA_FISICA.SEGUNDO_NOME || ' ' || CLIENTE_PESSOA_FISICA.ULTIMO_NOME AS "NOME COMPLETO",
       CONTA.NUMERO AS "NÚMERO DA CONTA"
  FROM CONTA,
       CLIENTE_PESSOA_FISICA
 WHERE CONTA.ID_CLIENTE = CLIENTE_PESSOA_FISICA.ID
 ORDER BY CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME
```

```
1  SELECT CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME || ' ' || CLIENTE_PESSOA_FISICA.SEGUNDO_NOME || ' ' || CLIENTE_PESSOA_FISICA.ULTIMO_NOME AS "NOME COMPLETO",
2    CONTA.NUMERO,
3    CONTA.ID_TIPO_CONTA,
4    TIPO_CONTA.NOME
5
6    FROM CONTA,
7      AGENCIA_BANCARIA,
8      CLIENTE_PESSOA_FISICA,
9      TIPO_CONTA
10   WHERE CONTA.ID_AGENCIA_BANCARIA = AGENCIA_BANCARIA.ID
11     AND CONTA.ID_CLIENTE = CLIENTE_PESSOA_FISICA.ID
12     AND CONTA.ID_TIPO_CONTA = TIPO_CONTA.ID
13     AND TIPO_CONTA.ID = 2
14
15   ORDER BY CLIENTE_PESSOA_FISICA.PRIMEIRO_NOME
```

Aplicação

The screenshot shows the homepage of the "Projeto Banco de Dados (Banco Financeiro)". The header is blue with the project name. Below it is a dark navigation bar with tabs: HOME (selected), CLIENTE, AGÊNCIA, BANCO, and ADMINISTRAÇÃO. A green button on the left contains a bar chart icon. The main title "Banco (Financeiro)" is displayed prominently. At the bottom, there is a row of four buttons labeled: CLIENTE, AGÊNCIA, BANCO, and ADMINISTRAÇÃO.

https://apex.oracle.com/pls/apex/banco_ev/r/projeto-banco/home?session=17227851655620



Obrigado!