|  |  |  |  |
| --- | --- | --- | --- |
| **Projeto:** | Sistema de gestão de Hotéis/Pousadas | **Versão:** | 1.0 |
| **Autores:** | João Pedro de Castro Laranjeira Rocha e Daniel do Carmo Nascimento | **Última Alteração:** | 20/08/2025 |

1. **TEMA:** Desenvolvimento de Sistemas aplicado à gestão de Hotéis/Pousadas
2. **DELIMITAÇÃO DO TEMA:** Sistema de Gestão de Hotéis/Pousadas, abrangendo desde a gestão de reservas, operações diárias até o controle financeiro e gerenciamento de pessoal.
3. **PROBLEMA:**
4. **HIPÓTESE:** A implementação de um banco de dados relacional, normalizado e centralizado permitirá a automação de processos-chave (reservas, faturamento, gestão de quartos), a redução de inconsistências e erros operacionais, e fornecerá uma base de dados íntegra para a geração de relatórios analíticos. Isso resultará em um aumento da eficiência operacional, melhor controle financeiro e, consequentemente, uma melhoria na experiência e satisfação do hóspede.
5. **JUSTIFICATIVA:** A competitividade no setor hoteleiro exige eficiência e uma experiência de cliente impecável. Um sistema de gestão centralizado é fundamental para alcançar esses objetivos. Este projeto se justifica pela necessidade de:

* Eficiência Operacional: Automatizar tarefas repetitivas, liberando a equipe para focar no atendimento ao cliente.
* Confiabilidade de Dados: Garantir que informações sobre disponibilidade de quartos, status de limpeza e faturamento sejam precisas e acessíveis em tempo real.
* Inteligência de Negócio: Permitir que gestores acessem relatórios sobre taxas de ocupação, receita por quarto, para embasar decisões estratégicas de precificação e marketing.

1. **OBJETIVO GERAL:** Projetar e modelar um banco de dados completo e integrado para um Sistema de Gestão de Hotéis/Pousadas, capaz de suportar de forma eficiente e segura todas as operações essenciais e avançadas de um hotel moderno.
2. **OBJETIVOS ESPECÍFICOS:** Modelar as entidades fundamentais para armazenar informações sobre reservas, hospedes, quartos, entre outros; Estruturar o processo de reservas, incluindo a lógica para tarifas dinâmicas baseadas em períodos; Desenvolver o esquema para gerenciar o ciclo de vida da estadia, do check-in ao check-out; Integrar um módulo financeiro para registrar o consumo de serviços, gerar faturas e processar pagamentos. Implementar um sistema de cargos e permissões para o controle de acesso dos funcionários. Permitir que funcionários com diferentes níveis de acesso realizem operações (ex: pessoal da faxina poder olhar quais quartos estão sujos, e assim que limpar, marcar que o quarto está disponível).
3. **DESCRIÇÃO DO SISTEMA:** Um potencial hóspede (ou um recepcionista) poderá consultar a disponibilidade de quartos para um período, com preços que variam conforme a temporada. Ao efetuar uma reserva, o quarto escolhido fica bloqueado para aquele período. Na chegada, o recepcionista realiza o check-in, transformando a reserva em uma estadia ativa e registrando todos os hóspedes. Durante a estadia, qualquer consumo extra (frigobar, lavanderia) é lançado na conta do quarto. No final, o sistema calcula o valor total, gera uma fatura detalhada e registra o pagamento. Gestores poderão acessar dashboards com relatórios de performance e a equipe de limpeza poderá atualizar o status dos quartos através do sistema.
4. **DESCRIÇÃO PARA GUIAR A MODELAGEM DE DADOS:**

Este sistema destina-se ao gerenciamento completo das operações de um hotel, abrangendo desde a infraestrutura de acomodações até a gestão de funcionários e faturamento.

**Hóspedes e Acomodações**

O hotel é composto por diversos quartos, cada um identificado por um número único e localizado em um andar específico. Os quartos possuem um status que reflete sua condição atual (ex: 'Disponível', 'Ocupado', 'Em Manutenção'). Cada quarto pertence a um único tipo de quarto. Os tipos de quarto são definidos por um nome (ex: 'Standard', 'Suíte Luxo'), uma descrição detalhada, sua capacidade máxima de pessoas e as comodidades oferecidas.

Para se hospedar, é necessário cadastrar os hóspedes. Cada hóspede será identificado por um código único, e seu CPF e e-mail também não poderão se repetir. Devemos registrar seu nome completo, data de nascimento, informações de contato (e-mail, telefone) e endereço.

**Reservas e Estadias**

O processo de hospedagem inicia-se com uma reserva. Uma reserva é sempre associada a um hóspede principal e registra a data em que foi feita, as datas previstas de check-in e check-out, e o número de adultos e crianças. As reservas possuem um status, como 'Confirmada', 'Pendente' ou 'Cancelada'. Uma reserva pode incluir um ou mais quartos.

Quando o hóspede chega ao hotel, a reserva é convertida em uma estadia. A estadia registra as datas reais de check-in e check-out e possui um status próprio ('Hospedado' ou 'Finalizada'). Uma estadia pode ser criada a partir de uma reserva pré-existente ou para hóspedes que chegam sem reserva ("walk-in"). É possível associar múltiplos hóspedes a uma mesma estadia.

**Serviços, Tarifas e Faturamento**

O hotel oferece serviços adicionais (como lavanderia, frigobar, etc.), cada um com nome, descrição e preço definidos. O consumo de serviços por parte de um hóspede durante sua estadia deve ser registrado, indicando qual serviço foi consumido, a quantidade e a data.

Os preços das diárias são dinâmicos e definidos por tarifas. Uma tarifa é o valor de um tipo de quarto para um determinado período do ano (ex: 'Alta Temporada', 'Baixa Temporada') e para um dia específico da semana. Cada período de tarifa possui um nome e datas de início e fim.

Ao final da estadia, é gerada uma fatura. Cada fatura está vinculada a uma única estadia e detalha os valores totais de diárias, serviços e impostos, resultando em um valor final. A fatura pode ter status como 'Pendente', 'Paga' ou 'Cancelada'. Um ou mais pagamentos podem ser realizados para quitar uma fatura, sendo necessário registrar o valor pago, o método e a data de cada pagamento.

**Funcionários e Controle de Acesso**

O sistema é operado por funcionários. Para cada funcionário, armazenamos dados pessoais e contratuais, como nome, data de admissão, salário e um status ('Ativo', 'Demitido'). Cada funcionário possui um nome de usuário e senha para acesso ao sistema.

Todo funcionário ocupa um cargo (ex: 'Recepcionista', 'Gerente', 'Camareira'). Cada cargo, por sua vez, está associado a um conjunto de permissões que definem as ações que o usuário pode realizar no sistema. Um cargo pode ter várias permissões, e uma permissão pode ser atribuída a vários cargos.

1. **MODELAGEM DE DADOS:**
   1. **MODELO RELACIONAL**

**Diagrama

O conteúdo gerado por IA pode estar incorreto.**

* 1. **VIEWS**

Texto

O conteúdo gerado por IA pode estar incorreto.

A VIEW\_QUARTOS\_OCUPADOS foi criada com o objetivo de permitir visualizar os quartos atualmente ocupados, em que andar estão localizados e o hóspede principal associado, além da respectiva data de checkout prevista.

A VIEW\_FUNCIONARIOS\_CARGOS foi criada com o objetivo de permitir visualizar informações sobre os funcionários cadastrados e seus respectivos cargos.

A VIEW\_CHECKOUTS\_PREVISTOS\_DIA\_CORRENTE foi criada com o objetivo de permitir visualizar as estadias que tem checkout previsto para o dia atual, mostrando o número do quarto e o hóspede principal relacionado à essa estadia.

**ANEXO ÚNICO**

**SCRIPT PARA IMPLANTAÇÃO DO PROJETO EM UM SGBD ORACLE 21C OU SUPERIOR**

-- Gerado por Oracle SQL Developer Data Modeler 24.3.1.351.0831

-- em: 2025-08-20 11:39:08 BRT

-- site: Oracle Database 21c

-- tipo: Oracle Database 21c

DROP USER SCH\_NEXUS CASCADE;

DROP USER SCH\_NEXUS\_IDENTITY CASCADE;

CREATE USER SCH\_NEXUS

IDENTIFIED BY sch\_nexus\_123

ACCOUNT UNLOCK

;

GRANT DBA TO SCH\_NEXUS

;

CREATE USER SCH\_NEXUS\_IDENTITY

IDENTIFIED BY sch\_nexus\_identity\_123

ACCOUNT UNLOCK

;

GRANT DBA TO SCH\_NEXUS\_IDENTITY

;

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE SCH\_NEXUS\_IDENTITY.CARGOS

(

ID\_CARGO NUMBER (10)

CONSTRAINT CARGO\_ID\_NOT\_NULL NOT NULL ,

NOM\_CARGO VARCHAR2 (50 CHAR)

CONSTRAINT CARGO\_NOME\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS\_IDENTITY.CARGOS TO SCH\_NEXUS\_IDENTITY WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.CARGOS

ADD CONSTRAINT CARGOS\_PK PRIMARY KEY ( ID\_CARGO ) ;

ALTER TABLE SCH\_NEXUS\_IDENTITY.CARGOS

ADD CONSTRAINT CARGOS\_NOME\_UN UNIQUE ( NOM\_CARGO ) ;

CREATE TABLE SCH\_NEXUS\_IDENTITY.CARGOS\_PERMISSOES

(

ID\_CARGO NUMBER (10)

CONSTRAINT CARGOS\_PERMISSOES\_CARGO\_ID\_NOT\_NULL NOT NULL ,

ID\_PERMISSAO NUMBER (10)

CONSTRAINT CARGOS\_PERMISSOES\_PERMISSAO\_ID\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS\_IDENTITY.CARGOS\_PERMISSOES TO SCH\_NEXUS\_IDENTITY WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.CARGOS\_PERMISSOES

ADD CONSTRAINT CARGOS\_PERMISSOES\_PK PRIMARY KEY ( ID\_CARGO, ID\_PERMISSAO ) ;

CREATE TABLE SCH\_NEXUS.CONSUMO\_SERVICOS

(

ID\_CONSUMO NUMBER (10)

CONSTRAINT CONSUMO\_SERVICOS\_ID\_NOT\_NULL NOT NULL ,

ID\_ESTADIA NUMBER (10)

CONSTRAINT CONSUMO\_SERVICOS\_ID\_ESTADIA\_NOT\_NULL NOT NULL ,

ID\_SERVICOS NUMBER (10)

CONSTRAINT CONSUMO\_SERVICOS\_ID\_SERVICOS\_NOT\_NULL NOT NULL ,

ID\_QUARTO NUMBER (10)

CONSTRAINT CONSUMO\_SERVICOS\_ID\_QUARTO\_NOT\_NULL NOT NULL ,

QTD\_SERVICO INTEGER

CONSTRAINT CONSUMO\_SERVICOS\_QUANTIDADE\_NOT\_NULL NOT NULL ,

PRECO\_TOTAL NUMBER (2) ,

DAT\_CONSUMO DATE

CONSTRAINT CONSUMO\_SERVICOS\_DATA\_CONSUMO\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.CONSUMO\_SERVICOS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.CONSUMO\_SERVICOS

ADD CONSTRAINT CONSUMO\_SERVICOS\_PK PRIMARY KEY ( ID\_CONSUMO ) ;

CREATE TABLE SCH\_NEXUS.ESTADIA

(

ID\_ESTADIA NUMBER (10)

CONSTRAINT ESTADIA\_ID\_NOT\_NULL NOT NULL ,

ID\_RESERVA NUMBER (10) ,

DAT\_CHECKIN\_REAL DATE

CONSTRAINT ESTADIA\_DATA\_CHECKIN\_REAL\_NOT\_NULL NOT NULL ,

DAT\_CHECKOUT\_REAL DATE

CONSTRAINT ESTADIA\_DATA\_CHECKOUT\_REAL\_NOT\_NULL NOT NULL ,

STS\_ESTADIA VARCHAR2 (20 CHAR)

CONSTRAINT ESTADIA\_STATUS\_NOT\_NULL NOT NULL

)

LOGGING

;

ALTER TABLE SCH\_NEXUS.ESTADIA

ADD CONSTRAINT ESTADIA\_STATUS\_CK

CHECK (STS\_ESTADIA IN ('HOSPEDADO', 'FINALIZADA'))

;

ALTER TABLE SCH\_NEXUS.ESTADIA

ADD CONSTRAINT ESTADIA\_PK PRIMARY KEY ( ID\_ESTADIA ) ;

CREATE TABLE SCH\_NEXUS.FATURAS

(

ID\_FATURA NUMBER (10)

CONSTRAINT FATURA\_ID\_NOT\_NULL NOT NULL ,

ID\_ESTADIA NUMBER (10)

CONSTRAINT FATURA\_ID\_ESTADIA\_NOT\_NULL NOT NULL ,

DAT\_EMISSAO DATE

CONSTRAINT FATURA\_DATA\_EMISSAO\_NOT\_NULL NOT NULL ,

VAL\_TOTAL\_DIARIAS NUMBER (10,2)

CONSTRAINT FATURA\_TOTAL\_DIARIAS\_NOT\_NULL NOT NULL ,

VAL\_TOTAL\_SERVICOS NUMBER (10,2)

CONSTRAINT FATURA\_TOTAL\_SERVICOS\_NOT\_NULL NOT NULL ,

VAL\_IMPOSTOS NUMBER (10,2)

CONSTRAINT FATURA\_TOTAL\_IMPOSTOS\_NOT\_NULL NOT NULL ,

VAL\_FINAL NUMBER (10,2)

CONSTRAINT FATURA\_VALOR\_FINAL\_NOT\_NULL NOT NULL ,

STS\_FATURA VARCHAR2 (20 CHAR)

CONSTRAINT FATURA\_STATUS\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.FATURAS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.FATURAS

ADD CONSTRAINT FATURAS\_STATUS\_CK

CHECK (STS\_FATURA IN ('PENDENTE', 'PAGA', 'PARCIALMENTE\_PAGA', 'CANCELADA'))

;

ALTER TABLE SCH\_NEXUS.FATURAS

ADD CONSTRAINT FATURAS\_PK PRIMARY KEY ( ID\_FATURA ) ;

CREATE TABLE SCH\_NEXUS\_IDENTITY.FUNCIONARIO

(

ID\_FUNCIONARIO NUMBER (10)

CONSTRAINT FUNCIONARIO\_ID\_NOT\_NULL NOT NULL ,

ID\_CARGO NUMBER (10)

CONSTRAINT FUNCIONARIO\_CARGO\_ID\_NOT\_NULL NOT NULL ,

NOM\_FUNCIONARIO VARCHAR2 (100 CHAR)

CONSTRAINT FUNCIONARIO\_NOME\_NOT\_NULL NOT NULL ,

DAT\_ADMISSAO DATE

CONSTRAINT FUNCIONARIO\_DATA\_ADMISSAO\_NOT\_NULL NOT NULL ,

DAT\_DEMISSAO DATE

CONSTRAINT FUNCIONARIO\_DATA\_DEMISSAO\_NOT\_NULL NOT NULL ,

SAL\_FUNIONARIO NUMBER (10,2)

CONSTRAINT FUNCIONARIO\_SALARIO\_NOT\_NULL NOT NULL ,

USER\_FUNCIONARIO VARCHAR2 (50 CHAR)

CONSTRAINT FUNCIONARIO\_USERNAME\_NOT\_NULL NOT NULL ,

PASS\_FUNCIONARIO VARCHAR2 (255 CHAR)

CONSTRAINT FUNCIONARIO\_PASSWORD\_NOT\_NULL NOT NULL ,

STS\_FUNCIONARIO VARCHAR2 (30 CHAR)

CONSTRAINT FUNCIONARIO\_STATUS\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS\_IDENTITY.FUNCIONARIO TO SCH\_NEXUS\_IDENTITY WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.FUNCIONARIO

ADD CONSTRAINT FUNCIONARIO\_CK\_1

CHECK (STS\_FUNCIONARIO IN ('ATIVO','DEMITIDO','AFASTADO'))

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.FUNCIONARIO

ADD CONSTRAINT FUNCIONARIO\_PK PRIMARY KEY ( ID\_FUNCIONARIO ) ;

ALTER TABLE SCH\_NEXUS\_IDENTITY.FUNCIONARIO

ADD CONSTRAINT FUNCIONARIO\_USERNAME\_UN UNIQUE ( USER\_FUNCIONARIO ) ;

CREATE TABLE SCH\_NEXUS.HOSPEDE

(

ID\_HOS NUMBER (10)

CONSTRAINT HOSPEDE\_ID\_NOT\_NULL NOT NULL ,

PNM\_HOS VARCHAR2 (50 CHAR)

CONSTRAINT HOSPEDE\_PNOME\_NOT\_NULL NOT NULL ,

UNM\_HOS VARCHAR2 (50 CHAR)

CONSTRAINT HOSPEDE\_UNOME\_NOT\_NULL NOT NULL ,

DAT\_NASCIMENTO\_HOS DATE

CONSTRAINT HOSPEDE\_DATA\_NASCIMENTO\_NOT\_NULL NOT NULL ,

CPF\_HOS CHAR (11 CHAR)

CONSTRAINT HOSPEDE\_CPF\_NOT\_NULL NOT NULL ,

EMAIL\_HOS VARCHAR2 (50 CHAR)

CONSTRAINT HOSPEDE\_EMAIL\_NOT\_NULL NOT NULL ,

TEL\_HOS VARCHAR2 (20 CHAR)

CONSTRAINT HOSPEDE\_TELEFONE\_NOT\_NULL NOT NULL ,

PAIS\_HOS VARCHAR2 (50 CHAR)

CONSTRAINT PAIS\_HOSPEDE\_NOT\_NULL NOT NULL ,

CIDADE\_HOS VARCHAR2 (50 CHAR)

CONSTRAINT HOSPEDE\_CIDADE\_NOT\_NULL NOT NULL ,

END\_HOSPEDE VARCHAR2 (100 CHAR)

CONSTRAINT HOSPEDE\_ENDERECO\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.HOSPEDE TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.HOSPEDE

ADD CONSTRAINT HOSPEDE\_PK PRIMARY KEY ( ID\_HOS ) ;

ALTER TABLE SCH\_NEXUS.HOSPEDE

ADD CONSTRAINT HOSPEDE\_CPF\_UN UNIQUE ( CPF\_HOS ) ;

ALTER TABLE SCH\_NEXUS.HOSPEDE

ADD CONSTRAINT HOSPEDE\_EMAIL\_UN UNIQUE ( EMAIL\_HOS ) ;

CREATE TABLE SCH\_NEXUS.HOSPEDES\_ESTADIA

(

ID\_ESTADIA NUMBER (10)

CONSTRAINT HOSPEDES\_ESTADIA\_ID\_ESTADIA\_NOT\_NULL NOT NULL ,

ID\_HOSPEDE NUMBER (10)

CONSTRAINT HOSPEDES\_ESTADIA\_ID\_HOSPEDE\_NOT\_NULL NOT NULL ,

ID\_QUARTO NUMBER (10)

CONSTRAINT HOSPEDES\_ESTADIA\_ID\_QUARTO\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.HOSPEDES\_ESTADIA TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.HOSPEDES\_ESTADIA

ADD CONSTRAINT HOSPEDES\_ESTADIA\_PK PRIMARY KEY ( ID\_ESTADIA, ID\_HOSPEDE, ID\_QUARTO ) ;

CREATE TABLE SCH\_NEXUS.PAGAMENTOS

(

ID\_PAGAMENTO NUMBER (10)

CONSTRAINT PAGAMENTO\_ID\_NOT\_NULL NOT NULL ,

ID\_FATURA NUMBER

CONSTRAINT PAGAMENTO\_ID\_FATURA\_NOT\_NULL NOT NULL ,

VALOR\_PAGO NUMBER (10,2)

CONSTRAINT PAGAMENTO\_VALOR\_PAGO\_NOT\_NULL NOT NULL ,

METODO\_PAGAMENTO VARCHAR2 (20 CHAR)

CONSTRAINT PAGAMENTO\_METODO\_PAGAMENTO\_NOT\_NULL NOT NULL ,

DAT\_PAGAMENTO DATE

CONSTRAINT PAGAMENTO\_DATA\_PAGAMENTO\_NOT\_NULL NOT NULL ,

OBS\_PAGAMENTO VARCHAR2 (255 CHAR)

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.PAGAMENTOS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.PAGAMENTOS

ADD CONSTRAINT PAGAMENTOS\_PK PRIMARY KEY ( ID\_PAGAMENTO ) ;

CREATE TABLE SCH\_NEXUS.PERIODOS\_TARIFA

(

ID\_PERIODO NUMBER (10)

CONSTRAINT PERIODO\_TARIFA\_ID\_NOT\_NULL NOT NULL ,

NOME\_PERIODO VARCHAR2 (50 CHAR)

CONSTRAINT TARIFA\_PERIODO\_NOME\_NOT\_NULL NOT NULL ,

DAT\_INICIO DATE

CONSTRAINT PERIODO\_TARIFA\_DATA\_INICIO\_NOT\_NULL NOT NULL ,

DAT\_FIM DATE

CONSTRAINT PERIODO\_TARIFA\_DATA\_FIM\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.PERIODOS\_TARIFA TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.PERIODOS\_TARIFA

ADD CONSTRAINT PERIODOS\_TARIFA\_PK PRIMARY KEY ( ID\_PERIODO ) ;

CREATE TABLE SCH\_NEXUS\_IDENTITY.PERMISSOES

(

ID\_PEMISSAO NUMBER (10)

CONSTRAINT PERMISSAO\_ID\_NOT\_NULL NOT NULL ,

NOME\_PERMISSAO VARCHAR2 (50 CHAR)

CONSTRAINT PERMISSAO\_NOME\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS\_IDENTITY.PERMISSOES TO SCH\_NEXUS\_IDENTITY WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.PERMISSOES

ADD CONSTRAINT PERMISSOES\_PK PRIMARY KEY ( ID\_PEMISSAO ) ;

CREATE TABLE SCH\_NEXUS.QUARTOS

(

ID\_QRT NUMBER (10) NOT NULL ,

NUM\_QRT VARCHAR2 (10 CHAR)

CONSTRAINT QUARTO\_NUMERO\_NOT\_NULL NOT NULL ,

ID\_TP\_QRT NUMBER (10) NOT NULL ,

AND\_QRT INTEGER

CONSTRAINT QUARTO\_ANDAR\_NOT\_NULL NOT NULL ,

STS\_QRT VARCHAR2 (25 CHAR)

CONSTRAINT QUARTO\_STATUS\_NOT\_NULL NOT NULL ,

OBS\_QRT VARCHAR2 (255 CHAR)

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.QUARTOS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.QUARTOS

ADD CONSTRAINT QUARTOS\_STATUS\_CK

CHECK (STS\_QRT IN ('DISPONIVEL', 'OCUPADO', 'LIMPEZA\_PENDENTE', 'EM\_MANUTENCAO','BLOQUEADO'))

;

ALTER TABLE SCH\_NEXUS.QUARTOS

ADD CONSTRAINT QUARTOS\_PK PRIMARY KEY ( ID\_QRT ) ;

ALTER TABLE SCH\_NEXUS.QUARTOS

ADD CONSTRAINT QUARTOS\_NUMERO\_UNIQUE UNIQUE ( NUM\_QRT ) ;

CREATE TABLE SCH\_NEXUS.QUARTOS\_RESERVADOS

(

ID\_RESERVA NUMBER (10)

CONSTRAINT RESERVA\_QUARTO\_ID\_RESERVA\_NOT\_NULL NOT NULL ,

ID\_QUARTO NUMBER (10)

CONSTRAINT RESERVA\_QUARTO\_ID\_QUARTO\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.QUARTOS\_RESERVADOS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.QUARTOS\_RESERVADOS

ADD CONSTRAINT RESERVA\_QUARTOS\_PK PRIMARY KEY ( ID\_RESERVA, ID\_QUARTO ) ;

CREATE TABLE SCH\_NEXUS.RESERVAS

(

ID\_RESERVA NUMBER (10)

CONSTRAINT RESERVA\_ID\_NOT\_NULL NOT NULL ,

ID\_HOSPEDE\_PRINCIPAL NUMBER (10)

CONSTRAINT RESERVA\_ID\_HOSPEDE\_PRINCIPAL\_NOT\_NULL NOT NULL ,

DAT\_RESERVA DATE

CONSTRAINT RESERVA\_DATA\_RESERVA\_NOT\_NULL NOT NULL ,

DAT\_CHECKIN\_PREVISTA DATE

CONSTRAINT RESERVA\_DATA\_CHECKIN\_PREVISTA NOT NULL ,

DAT\_CHECKOUT\_PREVISTA DATE

CONSTRAINT RESERVA\_DATA\_CHECKOUT\_PREVISTA NOT NULL ,

NUM\_ADULTOS INTEGER

CONSTRAINT RESERVA\_NUMERO\_ADULTOS\_NOT\_NULL NOT NULL ,

NUM\_CRIANCAS INTEGER ,

STS\_RESERVA VARCHAR2 (20 CHAR)

CONSTRAINT RESERVA\_STATUS\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.RESERVAS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.RESERVAS

ADD CONSTRAINT RESERVAS\_STATUS\_CK

CHECK (STS\_RESERVA IN ('CONFIRMADA', 'PENDENTE', 'CANCELADA', 'NOSHOW'))

;

ALTER TABLE SCH\_NEXUS.RESERVAS

ADD CONSTRAINT RESERVAS\_PK PRIMARY KEY ( ID\_RESERVA ) ;

CREATE TABLE SCH\_NEXUS.SERVICOS

(

ID\_SERVICO NUMBER (10)

CONSTRAINT SERVICO\_ID\_NOT\_NULL NOT NULL ,

NOME\_SERVICO VARCHAR2 (50 BYTE)

CONSTRAINT SERVICO\_NOME\_NOT\_NULL NOT NULL ,

DES\_SERVICO VARCHAR2 (500 CHAR)

CONSTRAINT SERVICO\_DESCRICAO\_NOT\_NULL NOT NULL ,

PRECO\_SERVICO NUMBER (10,2)

CONSTRAINT SERVICO\_PRECO\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.SERVICOS TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.SERVICOS

ADD CONSTRAINT SERVICOS\_PK PRIMARY KEY ( ID\_SERVICO ) ;

CREATE TABLE SCH\_NEXUS.TARIFA

(

ID\_TARIFA NUMBER (10)

CONSTRAINT TARIFA\_ID\_NOT\_NULL NOT NULL ,

ID\_TIPO\_QUARTO NUMBER (10)

CONSTRAINT TARIFA\_ID\_TIPO\_QUARTO\_NOT\_NULL NOT NULL ,

ID\_PERIODO\_TARIFA NUMBER (10) NOT NULL ,

PREC\_DIARIA NUMBER (10,2)

CONSTRAINT TARIFA\_PRECO\_DIARIA\_NOT\_NULL NOT NULL ,

DIA\_SEMANA VARCHAR2 (15 CHAR)

CONSTRAINT TARIFA\_DIA\_SEMANA\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.TARIFA TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.TARIFA

ADD CONSTRAINT TARIFA\_PK PRIMARY KEY ( ID\_TARIFA ) ;

CREATE TABLE SCH\_NEXUS.TIPOS\_QUARTO

(

ID\_TP\_QRT NUMBER (10)

CONSTRAINT NNC\_TIPOS\_QUARTO\_id\_tipo\_quarto NOT NULL ,

NOME\_TP\_QRT VARCHAR2 (50 CHAR)

CONSTRAINT TIPO\_QUARTO\_NOME\_NOT\_NULL NOT NULL ,

DESC\_TP\_QRT VARCHAR2 (500 CHAR)

CONSTRAINT TIPO\_QUARTO\_DESCRICAO\_NOT\_NULL NOT NULL ,

CAP\_MAXIMA\_TP\_QRT INTEGER

CONSTRAINT TIPO\_QUARTO\_CAPACIDADE\_MAXIMA\_NOT\_NULL NOT NULL ,

COMODIDADES\_TP\_QRT VARCHAR2 (100 CHAR)

CONSTRAINT TIPO\_QUARTO\_COMODIDADES\_NOT\_NULL NOT NULL

)

LOGGING

;

GRANT ALL PRIVILEGES

ON SCH\_NEXUS.TIPOS\_QUARTO TO SCH\_NEXUS WITH GRANT OPTION

;

ALTER TABLE SCH\_NEXUS.TIPOS\_QUARTO

ADD CONSTRAINT TIPOS\_QUARTO\_PK PRIMARY KEY ( ID\_TP\_QRT ) ;

ALTER TABLE SCH\_NEXUS.TIPOS\_QUARTO

ADD CONSTRAINT TIPOS\_QUARTO\_NOME\_UNIQUE UNIQUE ( NOME\_TP\_QRT ) ;

ALTER TABLE SCH\_NEXUS\_IDENTITY.CARGOS\_PERMISSOES

ADD CONSTRAINT CARGOS\_PERMISSOES\_CARGOS\_FK FOREIGN KEY

(

ID\_CARGO

)

REFERENCES SCH\_NEXUS\_IDENTITY.CARGOS

(

ID\_CARGO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.CARGOS\_PERMISSOES

ADD CONSTRAINT CARGOS\_PERMISSOES\_PERMISSOES\_FK FOREIGN KEY

(

ID\_PERMISSAO

)

REFERENCES SCH\_NEXUS\_IDENTITY.PERMISSOES

(

ID\_PEMISSAO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.CONSUMO\_SERVICOS

ADD CONSTRAINT CONSUMO\_SERVICOS\_ESTADIA\_FK FOREIGN KEY

(

ID\_ESTADIA

)

REFERENCES SCH\_NEXUS.ESTADIA

(

ID\_ESTADIA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.CONSUMO\_SERVICOS

ADD CONSTRAINT CONSUMO\_SERVICOS\_QUARTOS\_FK FOREIGN KEY

(

ID\_QUARTO

)

REFERENCES SCH\_NEXUS.QUARTOS

(

ID\_QRT

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.CONSUMO\_SERVICOS

ADD CONSTRAINT CONSUMO\_SERVICOS\_SERVICOS\_FK FOREIGN KEY

(

ID\_SERVICOS

)

REFERENCES SCH\_NEXUS.SERVICOS

(

ID\_SERVICO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.ESTADIA

ADD CONSTRAINT ESTADIA\_RESERVAS\_FK FOREIGN KEY

(

ID\_RESERVA

)

REFERENCES SCH\_NEXUS.RESERVAS

(

ID\_RESERVA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.FATURAS

ADD CONSTRAINT FATURAS\_ESTADIA\_FK FOREIGN KEY

(

ID\_ESTADIA

)

REFERENCES SCH\_NEXUS.ESTADIA

(

ID\_ESTADIA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.FUNCIONARIO

ADD CONSTRAINT FUNCIONARIO\_CARGOS\_FK FOREIGN KEY

(

ID\_CARGO

)

REFERENCES SCH\_NEXUS\_IDENTITY.CARGOS

(

ID\_CARGO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.HOSPEDES\_ESTADIA

ADD CONSTRAINT HOSPEDES\_ESTADIA\_ESTADIA\_FK FOREIGN KEY

(

ID\_ESTADIA

)

REFERENCES SCH\_NEXUS.ESTADIA

(

ID\_ESTADIA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.HOSPEDES\_ESTADIA

ADD CONSTRAINT HOSPEDES\_ESTADIA\_HOSPEDE\_FK FOREIGN KEY

(

ID\_HOSPEDE

)

REFERENCES SCH\_NEXUS.HOSPEDE

(

ID\_HOS

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.HOSPEDES\_ESTADIA

ADD CONSTRAINT HOSPEDES\_ESTADIA\_QUARTOS\_FK FOREIGN KEY

(

ID\_QUARTO

)

REFERENCES SCH\_NEXUS.QUARTOS

(

ID\_QRT

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.PAGAMENTOS

ADD CONSTRAINT PAGAMENTOS\_FATURAS\_FK FOREIGN KEY

(

ID\_FATURA

)

REFERENCES SCH\_NEXUS.FATURAS

(

ID\_FATURA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.QUARTOS\_RESERVADOS

ADD CONSTRAINT QUARTOS\_RESERVADOS\_ID\_QUARTOS\_FK FOREIGN KEY

(

ID\_QUARTO

)

REFERENCES SCH\_NEXUS.QUARTOS

(

ID\_QRT

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.QUARTOS\_RESERVADOS

ADD CONSTRAINT QUARTOS\_RESERVADOS\_ID\_RESERVA\_FK FOREIGN KEY

(

ID\_RESERVA

)

REFERENCES SCH\_NEXUS.RESERVAS

(

ID\_RESERVA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.QUARTOS

ADD CONSTRAINT QUARTOS\_TIPOS\_QUARTO\_FK FOREIGN KEY

(

ID\_TP\_QRT

)

REFERENCES SCH\_NEXUS.TIPOS\_QUARTO

(

ID\_TP\_QRT

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.RESERVAS

ADD CONSTRAINT RESERVAS\_HOSPEDE\_FK FOREIGN KEY

(

ID\_HOSPEDE\_PRINCIPAL

)

REFERENCES SCH\_NEXUS.HOSPEDE

(

ID\_HOS

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TARIFA

ADD CONSTRAINT TARIFA\_PERIODOS\_TARIFA\_FK FOREIGN KEY

(

ID\_PERIODO\_TARIFA

)

REFERENCES SCH\_NEXUS.PERIODOS\_TARIFA

(

ID\_PERIODO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TARIFA

ADD CONSTRAINT TARIFA\_TIPOS\_QUARTO\_FK FOREIGN KEY

(

ID\_TIPO\_QUARTO

)

REFERENCES SCH\_NEXUS.TIPOS\_QUARTO

(

ID\_TP\_QRT

)

NOT DEFERRABLE

;

CREATE OR REPLACE VIEW VIEW\_CHECKOUTS\_PREVISTOS\_DIA\_CORRENTE ( ID\_ESTADIA, NOME\_HOSPEDE, NUM\_QRT, DAT\_CHECKOUT\_PREVISTA ) AS

SELECT

e.ID\_ESTADIA,

h.PNM\_HOS || ' ' || h.UNM\_HOS AS NOME\_HOSPEDE,

q.NUM\_QRT,

r.DAT\_CHECKOUT\_PREVISTA

FROM

SCH\_NEXUS.ESTADIA e

JOIN

SCH\_NEXUS.RESERVAS r ON e.ID\_RESERVA = r.ID\_RESERVA

JOIN

SCH\_NEXUS.HOSPEDE h ON r.ID\_HOSPEDE\_PRINCIPAL = h.ID\_HOS

JOIN

SCH\_NEXUS.HOSPEDES\_ESTADIA he ON e.ID\_ESTADIA = he.ID\_ESTADIA

JOIN

SCH\_NEXUS.QUARTOS q ON he.ID\_QUARTO = q.ID\_QRT

WHERE

e.STS\_ESTADIA = 'HOSPEDADO'

AND TRUNC(r.DAT\_CHECKOUT\_PREVISTA) = TRUNC(SYSDATE)

;

CREATE OR REPLACE VIEW VIEW\_FUNCIONARIOS\_CARGOS ( ID\_FUNCIONARIO, NOM\_FUNCIONARIO, NOM\_CARGO, DAT\_ADMISSAO, STS\_FUNCIONARIO ) AS

SELECT

f.ID\_FUNCIONARIO,

f.NOM\_FUNCIONARIO,

c.NOM\_CARGO,

f.DAT\_ADMISSAO,

f.STS\_FUNCIONARIO

FROM

SCH\_NEXUS\_IDENTITY.FUNCIONARIO f

JOIN

SCH\_NEXUS\_IDENTITY.CARGOS c ON f.ID\_CARGO = c.ID\_CARGO

;

CREATE OR REPLACE VIEW VIEW\_QUARTOS\_OCUPADOS ( NUM\_QRT, NOME\_TP\_QRT, AND\_QRT, STS\_QRT, HOSPEDE\_PRINCIPAL, DAT\_CHECKOUT\_PREVISTA ) AS

SELECT

q.NUM\_QRT,

tq.NOME\_TP\_QRT,

q.AND\_QRT,

q.STS\_QRT,

h.PNM\_HOS || ' ' || h.UNM\_HOS AS HOSPEDE\_PRINCIPAL,

r.DAT\_CHECKOUT\_PREVISTA

FROM

SCH\_NEXUS.QUARTOS q

INNER JOIN

SCH\_NEXUS.TIPOS\_QUARTO tq ON q.ID\_TP\_QRT = tq.ID\_TP\_QRT

LEFT JOIN

SCH\_NEXUS.HOSPEDES\_ESTADIA he ON q.ID\_QRT = he.ID\_QUARTO

LEFT JOIN

SCH\_NEXUS.ESTADIA e ON he.ID\_ESTADIA = e.ID\_ESTADIA AND e.STS\_ESTADIA = 'HOSPEDADO'

LEFT JOIN

SCH\_NEXUS.RESERVAS r ON e.ID\_RESERVA = r.ID\_RESERVA

LEFT JOIN

SCH\_NEXUS.HOSPEDE h ON r.ID\_HOSPEDE\_PRINCIPAL = h.ID\_HOS

ORDER BY

q.NUM\_QRT

;

CREATE SEQUENCE SCH\_NEXUS\_IDENTITY.CARGOS\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS\_IDENTITY.CARGOS\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS\_IDENTITY.CARGOS

FOR EACH ROW

WHEN (NEW.ID\_CARGO IS NULL)

BEGIN

:NEW.ID\_CARGO := SCH\_NEXUS\_IDENTITY.CARGOS\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.CONSUMO\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.CONSUMO\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.CONSUMO\_SERVICOS

FOR EACH ROW

WHEN (NEW.ID\_CONSUMO IS NULL)

BEGIN

:NEW.ID\_CONSUMO := SCH\_NEXUS.CONSUMO\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.ESTADIA\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.ESTADIA\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.ESTADIA

FOR EACH ROW

WHEN (NEW.ID\_ESTADIA IS NULL)

BEGIN

:NEW.ID\_ESTADIA := SCH\_NEXUS.ESTADIA\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.FATURA\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.FATURA\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.FATURAS

FOR EACH ROW

WHEN (NEW.ID\_FATURA IS NULL)

BEGIN

:NEW.ID\_FATURA := SCH\_NEXUS.FATURA\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS\_IDENTITY.FUNCIONARIOS\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS\_IDENTITY.FUNCIONARIOS\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS\_IDENTITY.FUNCIONARIO

FOR EACH ROW

WHEN (NEW.ID\_FUNCIONARIO IS NULL)

BEGIN

:NEW.ID\_FUNCIONARIO := SCH\_NEXUS\_IDENTITY.FUNCIONARIOS\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.HOSPEDES\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.HOSPEDES\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.HOSPEDE

FOR EACH ROW

WHEN (NEW.ID\_HOS IS NULL)

BEGIN

:NEW.ID\_HOS := SCH\_NEXUS.HOSPEDES\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.PAGAMENTO\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.PAGAMENTO\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.PAGAMENTOS

FOR EACH ROW

WHEN (NEW.ID\_PAGAMENTO IS NULL)

BEGIN

:NEW.ID\_PAGAMENTO := SCH\_NEXUS.PAGAMENTO\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.PERIODO\_TARIFA\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.PERIODO\_TARIFA\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.PERIODOS\_TARIFA

FOR EACH ROW

WHEN (NEW.ID\_PERIODO IS NULL)

BEGIN

:NEW.ID\_PERIODO := SCH\_NEXUS.PERIODO\_TARIFA\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS\_IDENTITY.PERMISSOES\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS\_IDENTITY.PERMISSOES\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS\_IDENTITY.PERMISSOES

FOR EACH ROW

WHEN (NEW.ID\_PEMISSAO IS NULL)

BEGIN

:NEW.ID\_PEMISSAO := SCH\_NEXUS\_IDENTITY.PERMISSOES\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.QUARTO\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.QUARTO\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.QUARTOS

FOR EACH ROW

WHEN (NEW.ID\_QRT IS NULL)

BEGIN

:NEW.ID\_QRT := SCH\_NEXUS.QUARTO\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.RESERVAS\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.RESERVAS\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.RESERVAS

FOR EACH ROW

WHEN (NEW.ID\_RESERVA IS NULL)

BEGIN

:NEW.ID\_RESERVA := SCH\_NEXUS.RESERVAS\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.SERVICOS\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.SERVICOS\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.SERVICOS

FOR EACH ROW

WHEN (NEW.ID\_SERVICO IS NULL)

BEGIN

:NEW.ID\_SERVICO := SCH\_NEXUS.SERVICOS\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.TARIFA\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.TARIFA\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.TARIFA

FOR EACH ROW

WHEN (NEW.ID\_TARIFA IS NULL)

BEGIN

:NEW.ID\_TARIFA := SCH\_NEXUS.TARIFA\_ID\_SEQUENCE.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.TIPOS\_QUARTO\_ID\_SEQUENCE

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.TIPOS\_QUARTO\_ID\_TRIGGER

BEFORE INSERT ON SCH\_NEXUS.TIPOS\_QUARTO

FOR EACH ROW

WHEN (NEW.ID\_TP\_QRT IS NULL)

BEGIN

:NEW.ID\_TP\_QRT := SCH\_NEXUS.TIPOS\_QUARTO\_ID\_SEQUENCE.NEXTVAL;

END;

/

-- Relatório do Resumo do Oracle SQL Developer Data Modeler:

--

-- CREATE TABLE 17

-- CREATE INDEX 0

-- ALTER TABLE 46

-- CREATE VIEW 3

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 14

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 14

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 2

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0