

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

- TEMA:** Desenvolvimento de Sistemas aplicado à gestão de Hotéis/Pousadas
- 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.
- PROBLEMA:**
- 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, conseqüentemente, uma melhoria na experiência e satisfação do hóspede.
- 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.
- 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.
- OBJETIVOS ESPECÍFICOS:** Modelar as entidades fundamentais para armazenar informações sobre reservas, hóspedes, 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).
- 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.

9. 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.

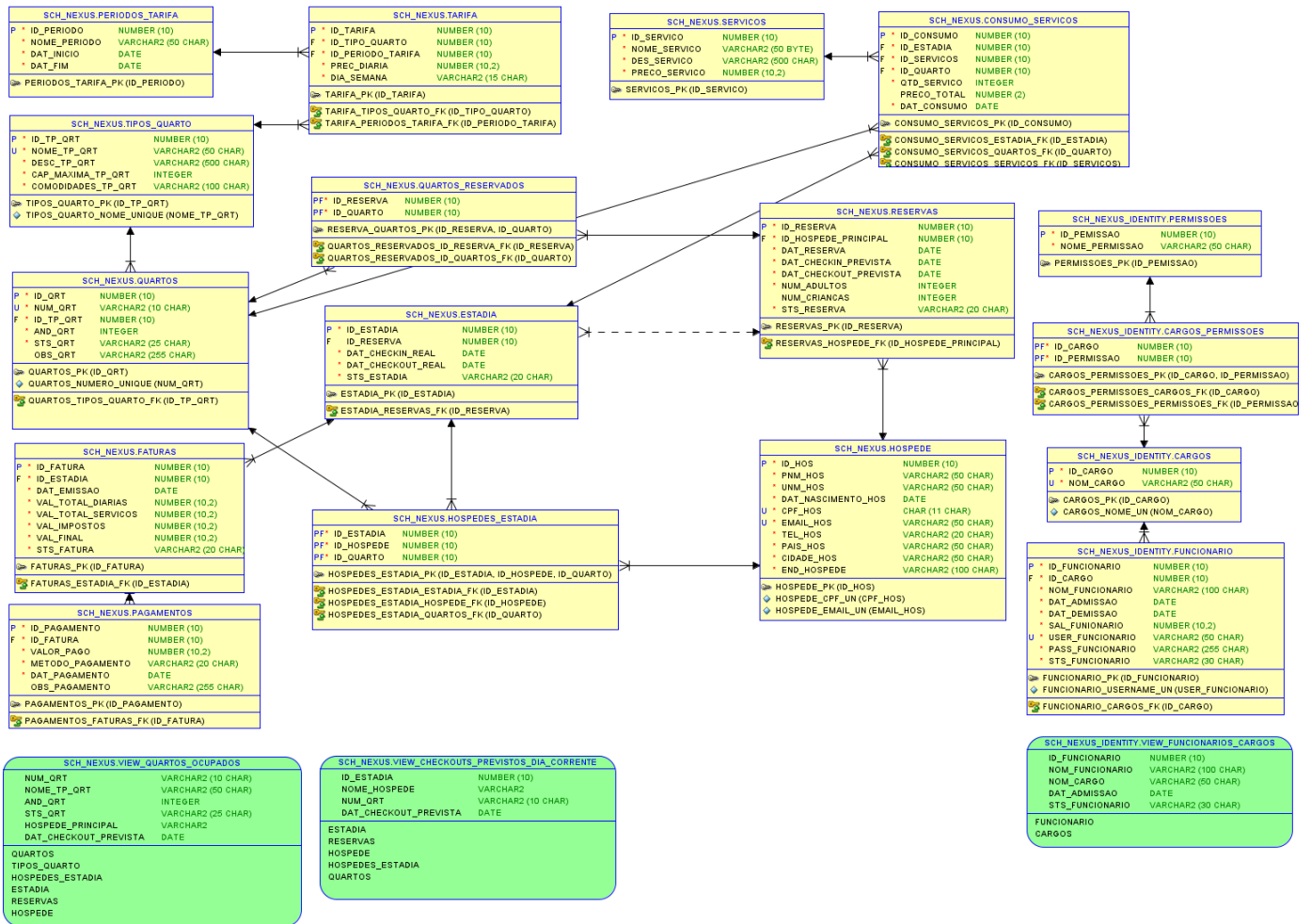
Ao final da estadia, é gerada uma fatura. Cada fatura está vinculada a uma única estadia e detalha os valores totais de diárias, serviços e impostos, resultando em um valor final. A fatura pode ter status como 'Pendente', 'Paga' ou 'Cancelada'. Um ou mais pagamentos podem ser realizados para quitar uma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

10.MODELAGEM DE DADOS:



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_FUNCIONARIO  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 SERVICOS_ID_NOT_NULL NOT NULL ,  
NOME_SERVICO VARCHAR2 (50 BYTE)  
CONSTRAINT SERVICOS_NOME_NOT_NULL NOT NULL ,  
DES_SERVICO VARCHAR2 (500 CHAR)  
CONSTRAINT SERVICOS_DESCRICAO_NOT_NULL NOT NULL ,  
PRECO_SERVICO NUMBER (10,2)  
CONSTRAINT SERVICOS_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

```