

<b>Projeto:</b>	Sistema de gestão de Hotéis/Pousadas	<b>Versão:</b>	1.0.3
<b>Autores:</b>	João Pedro de Castro Laranjeira Rocha e Daniel do Carmo Nascimento	<b>Última Alteração:</b>	20/10/2025

- 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.
- 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.

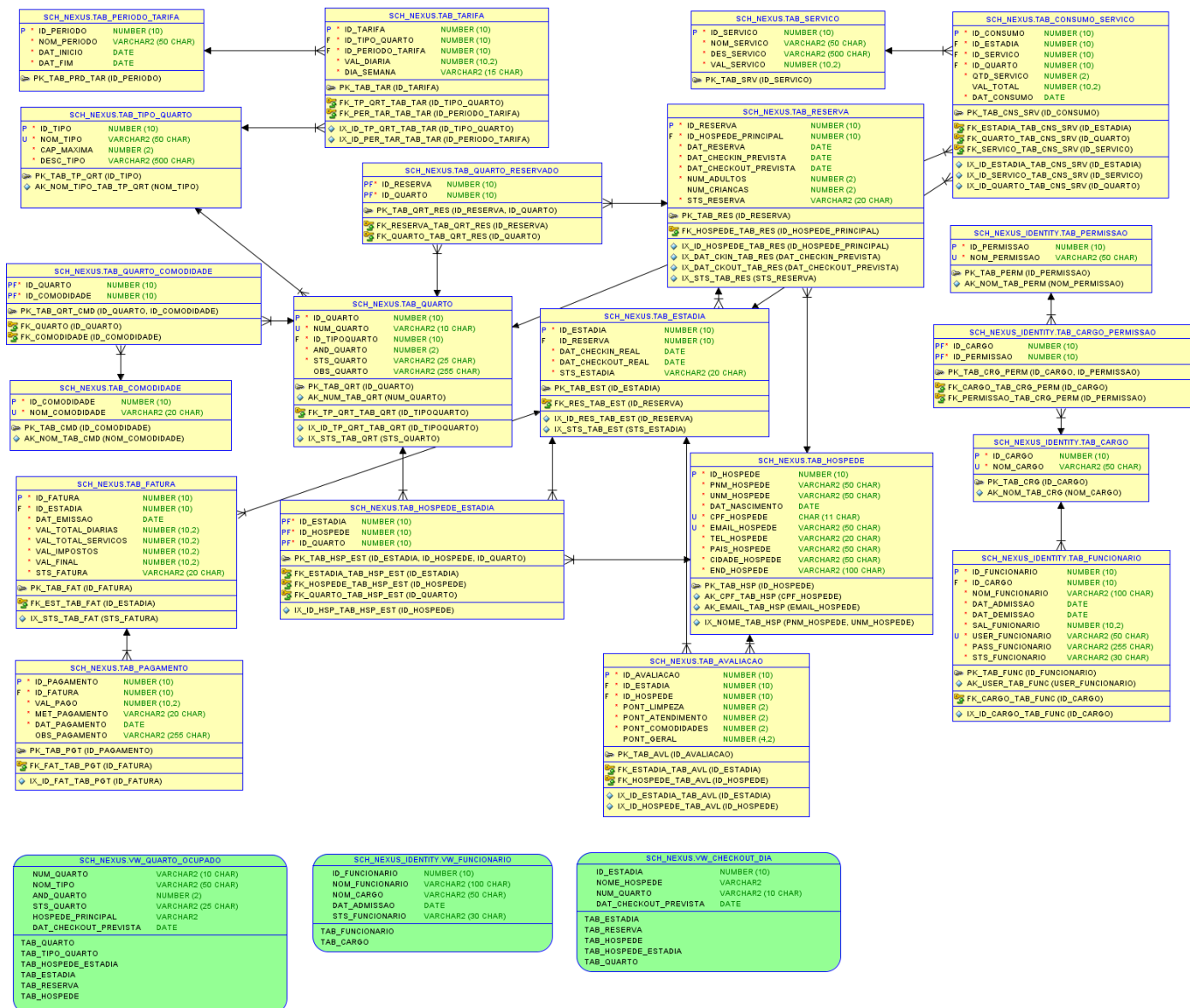
## **Funcionários e Controle de Acesso**

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.

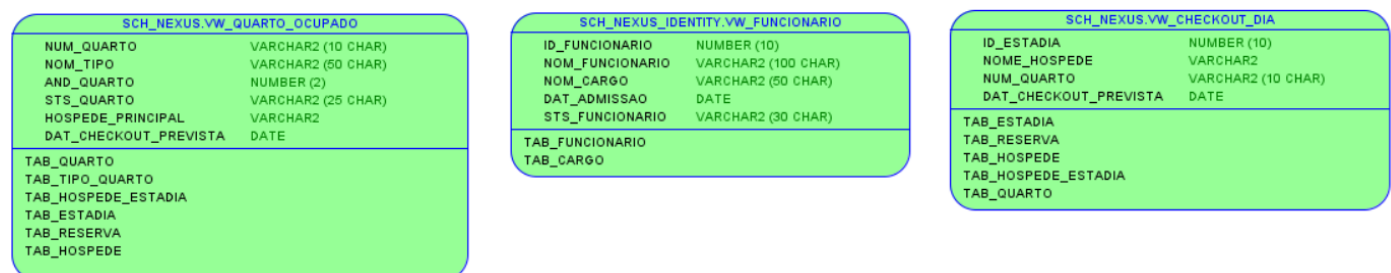
Todo funcionário ocupa um cargo (ex: 'Recepcionista', 'Gerente', 'Camareira'). Cada cargo, por sua vez, está associado a um conjunto de permissões que definem as ações que o usuário pode realizar no sistema. Um cargo pode ter várias permissões, e uma permissão pode ser atribuída a vários cargos.

## **10. MODELAGEM DE DADOS:**

### **10.1. MODELO RELACIONAL**



## 10.2. VIEWS



A VW\_QUARTO\_OCUPADO 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 VW\_FUNCIONARIO foi criada com o objetivo de permitir visualizar informações sobre os funcionários cadastrados e seus respectivos cargos.

A VW\_CHECKOUT\_DIA 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.

### 10.3. ÍNDICES

- **IX\_ID\_CARGO\_TAB\_FUNC:** Otimiza a junção entre TAB\_FUNCIONARIO e TAB\_CARGO, acelerando a busca de funcionários por cargo.
- **IX\_ID\_ESTADIA\_TAB\_AVL:** Acelera a busca de avaliações relacionadas a uma estadia específica.
- **IX\_ID\_HOSPEDE\_TAB\_AVL:** Permite encontrar rapidamente todas as avaliações feitas por um hóspede.
- **IX\_ID\_ESTADIA\_TAB\_CNS\_SRV, IX\_ID\_SERVICO\_TAB\_CNS\_SRV, IX\_ID\_QUARTO\_TAB\_CNS\_SRV:** Essenciais para otimizar consultas na tabela de consumo, permitindo filtrar rapidamente por estadia, serviço ou quarto.
- **IX\_ID\_RES\_TAB\_EST:** Torna a busca da estadia associada a uma reserva muito mais rápida.
- **IX\_ID\_HSP\_TAB\_HSP\_EST:** Otimiza a busca de todas as estadias de um hóspede específico.
- **IX\_ID\_FAT\_TAB\_PGT:** Acelera a consulta de pagamentos vinculados a uma fatura.
- **IX\_ID\_TP\_QRT\_TAB\_QRT:** Permite encontrar rapidamente todos os quartos de um determinado tipo (ex: 'todos os quartos Standard').
- **IX\_ID\_HOSPEDE\_TAB\_RES:** Otimiza a busca de todas as reservas feitas por um hóspede principal.
- **IX\_ID\_TP\_QRT\_TAB\_TAR e IX\_ID\_PER\_TAR\_TAB\_TAR:** Aceleram a busca de preços de diárias, consultas essenciais para o sistema de reservas e faturamento.
- **IX\_STS\_TAB\_EST, IX\_STS\_TAB\_FAT, IX\_STS\_TAB\_QRT, IX\_STS\_TAB\_RES:** Aceleram drasticamente as consultas mais comuns do sistema, como encontrar quartos DISPONÍVEIS, faturas PENDENTES, reservas CONFIRMADAS etc.
- **IX\_NOME\_TAB\_HSP:** Permite que a busca de hóspedes por nome e sobrenome seja instantânea, melhorando a performance da tela de pesquisa.
- **IX\_DAT\_CKIN\_TAB\_RES e IX\_DAT\_CKOUT\_TAB\_RES:** Fundamentais para acelerar buscas por intervalo de datas, como verificar disponibilidade, listar chegadas do dia ou prever a ocupação.

### 11. ESTIMATIVA ANUAL DE ÁREA DE ARMAZENAMENTO NECESSÁRIA PARA AS TABELAS:

TABELA	TAM REGISTRO	QTD REGISTROS	AREA (KB)
TAB_PERIODO_TARIFA	74	10	0,72265625
TAB_TIPO_QUARTO	562	10	5,48828125
TAB_QUARTO_COMODIDADE	20	800	15,625
TAB_COMODIDADE	30	25	0,732421875
TAB_FATURA	97	13.700	1297,753906
TAB_PAGAMENTO	312	15.000	4570,3125
TAB_TARIFA	55	50	2,685546875
TAB_QUARTO_RESERVADO	20	13.700	267,578125
TAB_QUARTO	312	100	30,46875
TAB_HOSPEDE_ESTADIA	30	20.500	600,5859375
TAB_SERVICO	570	30	16,69921875

TAB_RESERVA	65	13.000	825,1953125
TAB_ESTADIA	54	13.700	722,4609375
TAB_HOSPEDE	398	14.400	5596,875
TAB_AVALIACAO	40	4.100	160,15625
TAB_CONSUMO_SERVICO	48	27.400	1284,375
TAB_PERMISSAO	60	40	2,34375
TAB_CARGO_PERMISSAO	20	250	4,8828125
TAB_CARGO	60	20	1,171875
TAB_FUNCIONARIO	479	58	27,13085938
TOTAL	-	-	15,08 MB

## 12. SEGURANÇA DA INFORMAÇÃO:

### 12.1. PRIVILÉGIOS

ATIVIDADE	PRIVILÉGIO	OWNER	OBJETO
RL_SCH_NEXUS_NEXUS_READ	SELECT	SCH_NEXUS	TAB_AVALIACAO, TAB_COMODIDADE, TAB_CONSUMO_SERVICO, TAB_ESTADIA, TAB_FATURA, TAB_HOSPEDE, TAB_HOSPEDE_ESTADIA, TAB_PAGAMENTO, TAB_PERIODO_TARIFA, TAB_QUARTO, TAB_QUARTO_COMODIDADE, TAB_QUARTO_RESERVADO, TAB_RESERVA, TAB_SERVICO, TAB_TARIFA, TAB_TIPO_QUARTO
RL_SCH_NEXUS_REC EP_RW	SELECT, INSERT, UPDATE	SCH_NEXUS	TAB_CONSUMO_SERVICO, TAB_ESTADIA, TAB_FATURA, TAB_HOSPEDE, TAB_HOSPEDE_ESTADIA, TAB_PAGAMENTO, TAB_QUARTO_RESERVADO, TAB_RESERVA
RL_SCH_NEXUS_LIMPEZA_RW	SELECT	SCH_NEXUS	TAB_QUARTO
	UPDATE	SCH_NEXUS	STS_QUARTO (TAB_QUARTO)
RL_SCH_NEXUS_ADM_DADOS_RW	SELECT, INSERT, UPDATE, DELETE	SCH_NEXUS	TAB_AVALIACAO, TAB_COMODIDADE, TAB_CONSUMO_SERVICO, TAB_ESTADIA, TAB_FATURA, TAB_HOSPEDE, TAB_HOSPEDE_ESTADIA, TAB_PAGAMENTO, TAB_PERIODO_TARIFA, TAB_QUARTO, TAB_QUARTO_COMODIDADE, TAB_QUARTO_RESERVADO, TAB_RESERVA,

			TAB_SERVICO, TAB_TARIFA, TAB_TIPO_QUARTO
RL_SCH_NEXUS_CON SULTA_VW	SELECT	SCH_NEXUS	VW_CHECKOUT_DIA, VW_QUARTO_OCUPADO
RL_SCH_NEXUS_IDEN TITY_ID_FUNC_RW	SELECT, INSERT, UPDATE	SCH_NEXUS_ IDENTITY	TAB_FUNCIONARIO, TAB_CARGO, TAB_PERMISSAO, TAB_CARGO_PERMISSAO
RL_SCH_NEXUS_IDEN TITY_ID_READ	SELECT	SCH_NEXUS_ IDENTITY	TAB_CARGO, TAB_CARGO_PERMISSAO, TAB_FUNCIONARIO, TAB_PERMISSAO

## 12.2. ATRIBUIÇÕES

PAPEL	ATRIBUIÇÕES
GP_RECEPCIONISTA	RL_SCH_NEXUS_CONSULTA_VW RL_SCH_NEXUS_IDENTITY_ID_READ, RL_SCH_NEXUS_NEXUS_READ, RL_SCH_NEXUS_RECEP_RW
GP_LIMPEZA	RL_SCH_NEXUS_CONSULTA_VW, RL_SCH_NEXUS_LIMPEZA_RW
GP_GERENTE	RL_SCH_NEXUS_ADM_DADOS_RW, RL_SCH_NEXUS_CONSULTA_VW, RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW
GP_CLIENTE	RL_SCH_NEXUS_CONSULTA_VW

## 12.3. USUÁRIOS

NOME	USERNAME	PAPEL
Daniel MacLaren	DANIEL_RECEPCIONISTA	GP_RECEPCIONISTA
João Ferrari	JOAO_SERV_GERAIS	GP_LIMPEZA
Maria Mercedes	MARIA_GERENTE	GP_GERENTE
Davi Lotus	DAVI_CLIENTE	GP_CLIENTE

## 12.4. SCHEMA

ESQUEMA/USUÁRIO	PRIVILÉGIOS/ATRIBUIÇÕES
SCH_NEXUS	CONNECT
	RESOURCE
	CREATE VIEW
CARACTERÍSTICAS ADICIONAIS	
DEFAULT TABLESPACE TBS_NEXUS_DATA_001	
QUOTA UNLIMITED ON TBS_NEXUS_DATA_001	
QUOTA UNLIMITED ON TBS_NEXUS_INDEX_001	

ESQUEMA/USUÁRIO	PRIVILÉGIOS/ATRIBUIÇÕES
SCH_NEXUS_IDENTITY	CONNECT
	RESOURCE
	CREATE VIEW

CARACTERÍSTICAS ADICIONAIS
DEFAULT TABLESPACE TBS_NEXUS_IDENTITY_DATA_001
QUOTA UNLIMITED ON TBS_NEXUS_IDENTITY_DATA_001
QUOTA UNLIMITED ON TBS_NEXUS_IDENTITY_INDEX_001

## 12.5. POLÍTICA DE CRIAÇÃO DE SENHAS E UTILIZAÇÃO DE RECURSOS

### PROFILE: PRO\_PADRAO\_USUARIO

SENHAS		
Bloqueio de conta	FAILED_LOGIN_ATTEMPTS	5 tentativas
	PASSWORD_LOCK_TIME	UNLIMITED
Expiração e Vencimento da Senha	PASSWORD_LIFE_TIME	30 dias
	PASSWORD_GRACE_TIME	5 dias
Histórico de Senha (Quarentena)	PASSWORD_REUSE_TIME	30 dias
	PASSWORD_REUSE_MAX	-
Complexidade de Senha	PASSWORD_VERIFY_FUNCTION	DEFAULT

RECURSOS		
Conexão	SESSIONS_PER_USER	1 sessão
	CONNECT_TIME	300 minutos
	IDLE_TIME	30 minutos

## 13.

## 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: 2025-10-20 10:05:43 BRT

-- site: Oracle Database 21c

-- tipo: Oracle Database 21c

--- DROP DEFAULTS

--- USERS

DROP USER SCH\_NEXUS CASCADE;

DROP USER SCH\_NEXUS\_IDENTITY CASCADE;

DROP USER DAVI\_CLIENTE CASCADE;

DROP USER DANIEL\_RECEPCIONISTA CASCADE;

DROP USER MARIA\_GERENTE CASCADE;

DROP USER JOAO\_SERV\_GERAIS CASCADE;

--- PROFILES

DROP PROFILE PRO\_PADRAO\_USUARIO;

--- TABLESPACES

DROP TABLESPACE TBS\_NEXUS\_DATA\_001 INCLUDING CONTENTS AND DATAFILES CASCADE  
CONSTRAINTS;

DROP TABLESPACE TBS\_NEXUS\_IDENTITY\_DATA\_001 INCLUDING CONTENTS AND DATAFILES CASCADE  
CONSTRAINTS;

DROP TABLESPACE TBS\_NEXUS\_IDENTITY\_INDEX\_001 INCLUDING CONTENTS AND DATAFILES CASCADE  
CONSTRAINTS;

DROP TABLESPACE TBS\_NEXUS\_INDEX\_001 INCLUDING CONTENTS AND DATAFILES CASCADE  
CONSTRAINTS;

--- ROLES

DROP ROLE GP\_CLIENTE;

DROP ROLE GP\_GERENTE;

DROP ROLE GP\_LIMPEZA;

DROP ROLE GP\_RECEPCIONISTA;

DROP ROLE RL\_SCH\_NEXUS\_ADM\_DADOS\_RW;

DROP ROLE RL\_SCH\_NEXUS\_CONSULTA\_VW;

DROP ROLE RL\_SCH\_NEXUS\_LIMPEZA\_RW;

DROP ROLE RL\_SCH\_NEXUS\_NEXUS\_READ;

DROP ROLE RL\_SCH\_NEXUS\_RECEP\_RW;

DROP ROLE RL\_SCH\_NEXUS\_IDENTITY\_ID\_FUNC\_RW;



```
DROP ROLE RL_SCH_NEXUS_IDENTITY_ID_READ;
```

```
-- Criação de Profiles
```

```
CREATE PROFILE PRO_PADRAO_USUARIO LIMIT
```

```
    FAILED_LOGIN_ATTEMPTS 5
```

```
    PASSWORD_LOCK_TIME UNLIMITED
```

```
    PASSWORD_LIFE_TIME 30
```

```
    PASSWORD_GRACE_TIME 5
```

```
    PASSWORD_REUSE_TIME 30
```

```
    PASSWORD_VERIFY_FUNCTION DEFAULT
```

```
    SESSIONS_PER_USER 1
```

```
    CONNECT_TIME 300
```

```
    IDLE_TIME 30
```

```
;
```

```
CREATE SMALLFILE TABLESPACE TBS_NEXUS_DATA_001
```

```
    DATAFILE
```

```
        'DF_TBS_NEXUS_DATA_001' SIZE 15 M REUSE AUTOEXTEND ON NEXT 10 M MAXSIZE UNLIMITED
```

```
    LOGGING
```

```
    ONLINE
```

```
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1 M
```

```
    SEGMENT SPACE MANAGEMENT AUTO
```

```
    FLASHBACK ON
```

```
;
```

```
CREATE SMALLFILE TABLESPACE TBS_NEXUS_IDENTITY_DATA_001
```

```
    DATAFILE
```

```
        'DF_TBS_NEXUS_IDENTITY_DATA_001' SIZE 10 M REUSE AUTOEXTEND ON NEXT 5 M MAXSIZE  
UNLIMITED
```

```
    LOGGING
```

```
    ONLINE
```

```
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1 M
```

```
    SEGMENT SPACE MANAGEMENT AUTO
```

```
    FLASHBACK ON
```

```
;
```

CREATE SMALLFILE TABLESPACE TBS\_NEXUS\_IDENTITY\_INDEX\_001

DATAFILE

'DF\_TBS\_NEXUS\_IDENTITY\_INDEX\_001' SIZE 10 M REUSE AUTOEXTEND ON NEXT 5 M MAXSIZE  
UNLIMITED

LOGGING

ONLINE

EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1 M

SEGMENT SPACE MANAGEMENT AUTO

FLASHBACK ON

;

CREATE SMALLFILE TABLESPACE TBS\_NEXUS\_INDEX\_001

DATAFILE

'DF\_TBS\_NEXUS\_INDEX\_001' SIZE 15 M REUSE AUTOEXTEND ON NEXT 10 M MAXSIZE UNLIMITED

LOGGING

ONLINE

EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1 M

SEGMENT SPACE MANAGEMENT AUTO

FLASHBACK ON

;

CREATE ROLE RL\_SCH\_NEXUS\_ADM\_DADOS\_RW

NOT IDENTIFIED

;

CREATE ROLE RL\_SCH\_NEXUS\_CONSULTA\_VW

NOT IDENTIFIED

;

CREATE ROLE RL\_SCH\_NEXUS\_IDENTITY\_ID\_FUNC\_RW

NOT IDENTIFIED

;

CREATE ROLE RL\_SCH\_NEXUS\_IDENTITY\_ID\_READ

NOT IDENTIFIED

;

```
CREATE ROLE RL_SCH_NEXUS_LIMPEZA_RW  
    NOT IDENTIFIED  
;
```

```
CREATE ROLE RL_SCH_NEXUS_NEXUS_READ  
    NOT IDENTIFIED  
;
```

```
CREATE ROLE RL_SCH_NEXUS_RECEP_RW  
    NOT IDENTIFIED  
;
```

```
CREATE ROLE GP_CLIENTE  
    NOT IDENTIFIED  
;
```

```
GRANT RL_SCH_NEXUS_CONSULTA_VW TO GP_CLIENTE  
;
```

```
CREATE ROLE GP_GERENTE  
    NOT IDENTIFIED  
;
```

```
GRANT  
RL_SCH_NEXUS_CONSULTA_VW,RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW,RL_SCH_NEXUS_ADM_DADOS_  
RW TO GP_GERENTE  
;
```

```
CREATE ROLE GP_LIMPEZA  
    NOT IDENTIFIED  
;
```

```
GRANT RL_SCH_NEXUS_CONSULTA_VW,RL_SCH_NEXUS_LIMPEZA_RW TO GP_LIMPEZA  
;
```

```
CREATE ROLE GP_RECEPCIONISTA
```

NOT IDENTIFIED

;

GRANT

RL\_SCH\_NEXUS\_RECEP\_RW,RL\_SCH\_NEXUS\_CONSULTA\_VW,RL\_SCH\_NEXUS\_NEXUS\_READ,RL\_SCH\_NEXUS\_IDENTITY\_ID\_READ TO GP\_RECEPCIONISTA

;

CREATE USER DANIEL\_RECEPCIONISTA

IDENTIFIED BY daniel123

PROFILE PRO\_PADRAO\_USUARIO

ACCOUNT UNLOCK

;

GRANT GP\_RECEPCIONISTA TO DANIEL\_RECEPCIONISTA

;

CREATE USER DAVI\_CLIENTE

IDENTIFIED BY joao1234

PROFILE PRO\_PADRAO\_USUARIO

ACCOUNT UNLOCK

;

GRANT GP\_CLIENTE TO DAVI\_CLIENTE

;

CREATE USER JOAO\_SERV\_GERAIS

IDENTIFIED BY joao123

PROFILE PRO\_PADRAO\_USUARIO

ACCOUNT UNLOCK

;

GRANT GP\_LIMPEZA TO JOAO\_SERV\_GERAIS

;

CREATE USER MARIA\_GERENTE

IDENTIFIED BY maria\_gerente

PROFILE PRO\_PADRAO\_USUARIO

ACCOUNT UNLOCK

;

GRANT GP\_GERENTE TO MARIA\_GERENTE

;

CREATE USER SCH\_NEXUS

IDENTIFIED BY sch\_nexus\_123

DEFAULT TABLESPACE TBS\_NEXUS\_DATA\_001

QUOTA UNLIMITED ON TBS\_NEXUS\_INDEX\_001

QUOTA UNLIMITED ON TBS\_NEXUS\_DATA\_001

PASSWORD EXPIRE

ACCOUNT LOCK

;

GRANT CONNECT, CREATE VIEW, RESOURCE TO SCH\_NEXUS

;

CREATE USER SCH\_NEXUS\_IDENTITY

IDENTIFIED BY sch\_nexus\_identity\_123

DEFAULT TABLESPACE TBS\_NEXUS\_IDENTITY\_DATA\_001

QUOTA UNLIMITED ON TBS\_NEXUS\_IDENTITY\_DATA\_001

QUOTA UNLIMITED ON TBS\_NEXUS\_IDENTITY\_INDEX\_001

ACCOUNT UNLOCK

;

GRANT CONNECT, CREATE VIEW, RESOURCE TO SCH\_NEXUS\_IDENTITY

;

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

-- predefined type, no DDL - XMLTYPE

CREATE TABLE SCH\_NEXUS.TAB\_AVALIACAO

(

```

ID_AVALIACAO    NUMBER (10)
    CONSTRAINT NN_ID_TAB_AVL NOT NULL ,
ID_ESTADIA      NUMBER (10)
    CONSTRAINT NN_ID_EST_TAB_AVL NOT NULL ,
ID_HOSPEDE      NUMBER (10)
    CONSTRAINT NN_ID_HSP_TAB_AVL NOT NULL ,
PONT_LIMPEZA    NUMBER (2)
    CONSTRAINT NN_PONT_LIMP_TAB_AVL NOT NULL ,
PONT_ATENDIMENTO NUMBER (2)
    CONSTRAINT NN_PONT_ATEN_TAB_AVL NOT NULL ,
PONT_COMODIDADES NUMBER (2)
    CONSTRAINT NN_PONT_CMD_TAB_AVL NOT NULL ,
PONT_GERAL      NUMBER (4,2)
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;
CREATE INDEX SCH_NEXUS.IX_ID_ESTADIA_TAB_AVL ON SCH_NEXUS.TAB_AVALIACAO
(
ID_ESTADIA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
PCTINCREASE 0
MINEXTENTS 1

```

```

MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_ID_HOSPEDE_TAB_AVL ON SCH_NEXUS.TAB_AVALIACAO
(
ID_HOSPEDE ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT )
LOGGING
;

GRANT SELECT
ON SCH_NEXUS.TAB_AVALIACAO TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT DELETE, INSERT, SELECT, UPDATE
ON SCH_NEXUS.TAB_AVALIACAO TO RL_SCH_NEXUS_ADM_DADOS_RW
;

ALTER TABLE SCH_NEXUS.TAB_AVALIACAO
ADD CONSTRAINT PK_TAB_AVL PRIMARY KEY ( ID_AVALIACAO );

CREATE TABLE SCH_NEXUS_IDENTITY.TAB_CARGO
(
ID_CARGO NUMBER (10)
CONSTRAINT NN_ID_TAB_CRG NOT NULL ,
NOM_CARGO VARCHAR2 (50 CHAR)

```

```

        CONSTRAINT NN_NOM_TAB_CRG NOT NULL
    )
    PCTFREE 10
    PCTUSED 40
    TABLESPACE TBS_NEXUS_IDENTITY_DATA_001
    LOGGING
    STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT
    )
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS_IDENTITY.TAB_CARGO TO SCH_NEXUS_IDENTITY WITH GRANT OPTION
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS_IDENTITY.TAB_CARGO TO RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW
;

GRANT SELECT
    ON SCH_NEXUS_IDENTITY.TAB_CARGO TO RL_SCH_NEXUS_IDENTITY_ID_READ
;

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_CARGO
    ADD CONSTRAINT PK_TAB_CRG PRIMARY KEY ( ID_CARGO );

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_CARGO
    ADD CONSTRAINT AK_NOM_TAB_CRG UNIQUE ( NOM_CARGO );

CREATE TABLE SCH_NEXUS_IDENTITY.TAB_CARGO_PERMISSAO
(
    ID_CARGO    NUMBER (10)

```



```

        CONSTRAINT NN_ID_CRG_TAB_CRG_PERM NOT NULL ,
ID_PERMISSAO NUMBER (10)
        CONSTRAINT NN_ID_PERM_TAB_CRG_PERM NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_IDENTITY_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;

GRANT ALL PRIVILEGES
ON SCH_NEXUS_IDENTITY.TAB_CARGO_PERMISSAO TO SCH_NEXUS_IDENTITY WITH GRANT OPTION
;
GRANT INSERT, SELECT, UPDATE
ON SCH_NEXUS_IDENTITY.TAB_CARGO_PERMISSAO TO RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW
;
GRANT SELECT
ON SCH_NEXUS_IDENTITY.TAB_CARGO_PERMISSAO TO RL_SCH_NEXUS_IDENTITY_ID_READ
;

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_CARGO_PERMISSAO
ADD CONSTRAINT PK_TAB_CRG_PERM PRIMARY KEY ( ID_CARGO, ID_PERMISSAO ) ;

CREATE TABLE SCH_NEXUS.TAB_COMODIDADE
(
ID_COMODIDADE NUMBER (10)
CONSTRAINT NN_ID_TAB_CMD NOT NULL ,

```

```

NOM_COMODIDADE VARCHAR2 (20 CHAR)
    CONSTRAINT NN_NOM_TAB_CMD NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;

GRANT SELECT
    ON SCH_NEXUS.TAB_COMODIDADE TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT DELETE, INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_COMODIDADE TO RL_SCH_NEXUS_ADM_DADOS_RW
;

ALTER TABLE SCH_NEXUS.TAB_COMODIDADE
    ADD CONSTRAINT PK_TAB_CMD PRIMARY KEY ( ID_COMODIDADE );

ALTER TABLE SCH_NEXUS.TAB_COMODIDADE
    ADD CONSTRAINT AK_NOM_TAB_CMD UNIQUE ( NOM_COMODIDADE );

CREATE TABLE SCH_NEXUS.TAB_CONSUMO_SERVICO
(
    ID_CONSUMO NUMBER (10)
        CONSTRAINT NN_ID_TAB_CNS_SRV NOT NULL ,
    ID_ESTADIA NUMBER (10)
        CONSTRAINT NN_ID_EST_TAB_CNS_SRV NOT NULL ,
    ID_SERVICO NUMBER (10)

```

```

        CONSTRAINT NN_ID_SERV_TAB_CNS_SRV NOT NULL ,
ID_QUARTO  NUMBER (10)
        CONSTRAINT NN_ID_QRT_TAB_CNS_SRV NOT NULL ,
QTD_SERVICO NUMBER (2)
        CONSTRAINT NN_QTD_SERV_TAB_CNS_SRV NOT NULL ,
VAL_TOTAL  NUMBER (10,2) ,
DAT_CONSUMO DATE
        CONSTRAINT NN_DAT_CNS_TAB_CNS_SRV NOT NULL
    )
    PCTFREE 10
    PCTUSED 40
    TABLESPACE TBS_NEXUS_DATA_001
    LOGGING
    STORAGE (
        PCTINCREASE 0
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        FREELISTS 1
        FREELIST GROUPS 1
        BUFFER_POOL DEFAULT
    )
;

CREATE INDEX SCH_NEXUS.IX_ID_ESTADIA_TAB_CNS_SRV ON SCH_NEXUS.TAB_CONSUMO_SERVICO
(
    ID_ESTADIA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING

```

```

;
CREATE INDEX SCH_NEXUS.IX_ID_SERVICO_TAB_CNS_SRV ON SCH_NEXUS.TAB_CONSUMO_SERVICO
(
    ID_SERVICO ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_ID_QUARTO_TAB_CNS_SRV ON SCH_NEXUS.TAB_CONSUMO_SERVICO
(
    ID_QUARTO ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS.TAB_CONSUMO_SERVICO TO SCH_NEXUS WITH GRANT OPTION
;

```

GRANT SELECT

ON SCH\_NEXUS.TAB\_CONSUMO\_SERVICO TO RL\_SCH\_NEXUS\_NEXUS\_READ

;

GRANT INSERT, SELECT, UPDATE

ON SCH\_NEXUS.TAB\_CONSUMO\_SERVICO TO RL\_SCH\_NEXUS\_RECEP\_RW

;

GRANT DELETE, INSERT, SELECT, UPDATE

ON SCH\_NEXUS.TAB\_CONSUMO\_SERVICO TO RL\_SCH\_NEXUS\_ADM\_DADOS\_RW

;

ALTER TABLE SCH\_NEXUS.TAB\_CONSUMO\_SERVICO

ADD CONSTRAINT PK\_TAB\_CNS\_SRV PRIMARY KEY ( ID\_CONSUMO );

CREATE TABLE SCH\_NEXUS.TAB\_ESTADIA

(

ID\_ESTADIA     NUMBER (10)

CONSTRAINT NN\_ID\_TAB\_EST NOT NULL ,

ID\_RESERVA     NUMBER (10) ,

DAT\_CHECKIN\_REAL   DATE

CONSTRAINT NN\_DAT\_CHECKIN\_REAL\_TAB\_EST NOT NULL ,

DAT\_CHECKOUT\_REAL   DATE

CONSTRAINT NN\_DAT\_CHECKOUT\_REAL\_TAB\_EST NOT NULL ,

STS\_ESTADIA     VARCHAR2 (20 CHAR)

CONSTRAINT NN\_STS\_TAB\_EST NOT NULL

)

PCTFREE 10

PCTUSED 40

TABLESPACE TBS\_NEXUS\_DATA\_001

LOGGING

STORAGE (

PCTINCREASE 0

MINEXTENTS 1

MAXEXTENTS UNLIMITED

FREELISTS 1

FREELIST GROUPS 1

BUFFER\_POOL DEFAULT

```

)
;
CREATE INDEX SCH_NEXUS.IX_ID_RES_TAB_EST ON SCH_NEXUS.TAB_ESTADIA
(
    ID_RESERVA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;
CREATE INDEX SCH_NEXUS.IX_STS_TAB_EST ON SCH_NEXUS.TAB_ESTADIA
(
    STS_ESTADIA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;
GRANT SELECT
    ON SCH_NEXUS.TAB_ESTADIA TO RL_SCH_NEXUS_NEXUS_READ
;
GRANT INSERT, SELECT, UPDATE

```

```
ON SCH_NEXUS.TAB_ESTADIA TO RL_SCH_NEXUS_RECEP_RW
;
GRANT DELETE, INSERT, SELECT, UPDATE
ON SCH_NEXUS.TAB_ESTADIA TO RL_SCH_NEXUS_ADM_DADOS_RW
;
```

```
ALTER TABLE SCH_NEXUS.TAB_ESTADIA
ADD CONSTRAINT CK_STS_TAB_EST
CHECK (STS_ESTADIA IN ('HOSPEDADO', 'FINALIZADA'))
;
ALTER TABLE SCH_NEXUS.TAB_ESTADIA
ADD CONSTRAINT PK_TAB_EST PRIMARY KEY ( ID_ESTADIA );
```

```
CREATE TABLE SCH_NEXUS.TAB_FATURA
(
ID_FATURA      NUMBER (10)
CONSTRAINT NN_ID_TAB_FAT NOT NULL ,
ID_ESTADIA     NUMBER (10)
CONSTRAINT NN_ID_ESTADIA_TAB_FAT NOT NULL ,
DAT_EMISSAO    DATE
CONSTRAINT NN_DAT_EMISSAO_TAB_FAT NOT NULL ,
VAL_TOTAL_DIARIAS  NUMBER (10,2)
CONSTRAINT NN_VAL_TOT_DIARIAS_TAB_FAT NOT NULL ,
VAL_TOTAL_SERVICOS  NUMBER (10,2)
CONSTRAINT NN_VAL_TOT_SERVI_TAB_FAT NOT NULL ,
VAL_IMPOSTOS     NUMBER (10,2)
CONSTRAINT NN_VAL_IMPOSTOS_TAB_FAT NOT NULL ,
VAL_FINAL       NUMBER (10,2)
CONSTRAINT NN_VAL_FINAL_TAB_FAT NOT NULL ,
STS_FATURA      VARCHAR2 (20 CHAR)
CONSTRAINT NN_STS_TAB_FAT NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
```

```

        STORAGE (
        PCTINCREASE 0
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        FREELISTS 1
        FREELIST GROUPS 1
        BUFFER_POOL DEFAULT
    )
;

CREATE INDEX SCH_NEXUS.IX_STS_TAB_FAT ON SCH_NEXUS.TAB_FATURA
(
    STS_FATURA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS.TAB_FATURA TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
    ON SCH_NEXUS.TAB_FATURA TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_FATURA TO RL_SCH_NEXUS_RECEP_RW
;

GRANT DELETE, INSERT, SELECT, UPDATE

```



ON SCH\_NEXUS.TAB\_FATURA TO RL\_SCH\_NEXUS\_ADM\_DADOS\_RW

;

ALTER TABLE SCH\_NEXUS.TAB\_FATURA

ADD CONSTRAINT CK\_STS\_TAB\_FAT

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

;

ALTER TABLE SCH\_NEXUS.TAB\_FATURA

ADD CONSTRAINT PK\_TAB\_FAT PRIMARY KEY ( ID\_FATURA ) ;

CREATE TABLE SCH\_NEXUS\_IDENTITY.TAB\_FUNCIONARIO

(

ID\_FUNCIONARIO NUMBER (10)

CONSTRAINT NN\_ID\_TAB\_FUNC NOT NULL ,

ID\_CARGO NUMBER (10)

CONSTRAINT NN\_ID\_CRG\_TAB\_FUNC NOT NULL ,

NOM\_FUNCIONARIO VARCHAR2 (100 CHAR)

CONSTRAINT NN\_NOM\_TAB\_FUNC NOT NULL ,

DAT\_ADMISSAO DATE

CONSTRAINT NN\_DAT\_ADMISSAO\_TAB\_FUNC NOT NULL ,

DAT\_DEMISSAO DATE

CONSTRAINT NN\_DAT\_DEMISSAO\_TAB\_FUNC NOT NULL ,

SAL\_FUNCIONARIO NUMBER (10,2)

CONSTRAINT NN\_SAL\_TAB\_FUNC NOT NULL ,

USER\_FUNCIONARIO VARCHAR2 (50 CHAR)

CONSTRAINT NN\_USER\_TAB\_FUNC NOT NULL ,

PASS\_FUNCIONARIO VARCHAR2 (255 CHAR)

CONSTRAINT NN\_PASS\_TAB\_FUNC NOT NULL ,

STS\_FUNCIONARIO VARCHAR2 (30 CHAR)

CONSTRAINT NN\_STS\_TAB\_FUNC NOT NULL

)

PCTFREE 10

PCTUSED 40

TABLESPACE TBS\_NEXUS\_IDENTITY\_DATA\_001

LOGGING

STORAGE (

```

PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;

CREATE INDEX SCH_NEXUS_IDENTITY.IX_ID_CARGO_TAB_FUNC ON
SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO
(
ID_CARGO ASC
)
TABLESPACE TBS_NEXUS_IDENTITY_INDEX_001
PCTFREE 10
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
ON SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO TO SCH_NEXUS_IDENTITY WITH GRANT OPTION
;

GRANT INSERT, SELECT, UPDATE
ON SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO TO RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW
;

GRANT SELECT
ON SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO TO RL_SCH_NEXUS_IDENTITY_ID_READ
;

```

```

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO
ADD CONSTRAINT CK_STS_TAB_FUNC
CHECK (STS_FUNCIONARIO IN ('ATIVO','DEMITIDO','AFASTADO'))
;

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO
ADD CONSTRAINT PK_TAB_FUNC PRIMARY KEY ( ID_FUNCIONARIO ) ;

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_FUNCIONARIO
ADD CONSTRAINT AK_USER_TAB_FUNC UNIQUE ( USER_FUNCIONARIO ) ;

CREATE TABLE SCH_NEXUS.TAB_HOSPEDE
(
ID_HOSPEDE    NUMBER (10)
CONSTRAINT NN_ID_TAB_HSP NOT NULL ,
PNM_HOSPEDE   VARCHAR2 (50 CHAR)
CONSTRAINT NN_PNM_TAB_HSP NOT NULL ,
UNM_HOSPEDE   VARCHAR2 (50 CHAR)
CONSTRAINT NN_UNM_TAB_HSP NOT NULL ,
DAT_NASCIMENTO DATE
CONSTRAINT NN_DAT_NAS_TAB_HSP NOT NULL ,
CPF_HOSPEDE   CHAR (11 CHAR)
CONSTRAINT NN_CPF_TAB_HSP NOT NULL ,
EMAIL_HOSPEDE VARCHAR2 (50 CHAR)
CONSTRAINT NN_EMAIL_TAB_HSP NOT NULL ,
TEL_HOSPEDE   VARCHAR2 (20 CHAR)
CONSTRAINT NN_TEL_TAB_HSP NOT NULL ,
PAIS_HOSPEDE  VARCHAR2 (50 CHAR)
CONSTRAINT NN_PAIS_TAB_HSP NOT NULL ,
CIDADE_HOSPEDE VARCHAR2 (50 CHAR)
CONSTRAINT NN_CID_TAB_HSP NOT NULL ,
END_HOSPEDE   VARCHAR2 (100 CHAR)
CONSTRAINT NN_END_TAB_HSP NOT NULL
)

PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001

```

```

    LOGGING
    STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT
    )
;

CREATE INDEX SCH_NEXUS.IX_NOME_TAB_HSP ON SCH_NEXUS.TAB_HOSPEDE
(
    PNM_HOSPEDE ASC ,
    UNM_HOSPEDE ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS.TAB_HOSPEDE TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
    ON SCH_NEXUS.TAB_HOSPEDE TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_HOSPEDE TO RL_SCH_NEXUS_RECEP_RW

```

```
;  
  
GRANT DELETE, INSERT, SELECT, UPDATE  
    ON SCH_NEXUS.TAB_HOSPEDE TO RL_SCH_NEXUS_ADM_DADOS_RW  
;  

```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE  
    ADD CONSTRAINT PK_TAB_HSP PRIMARY KEY ( ID_HOSPEDE );
```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE  
    ADD CONSTRAINT AK_CPF_TAB_HSP UNIQUE ( CPF_HOSPEDE );
```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE  
    ADD CONSTRAINT AK_EMAIL_TAB_HSP UNIQUE ( EMAIL_HOSPEDE );
```

```
CREATE TABLE SCH_NEXUS.TAB_HOSPEDE_ESTADIA  
(  
    ID_ESTADIA NUMBER (10)  
        CONSTRAINT NN_ID_EST_TAB_HSP_EST NOT NULL ,  
    ID_HOSPEDE NUMBER (10)  
        CONSTRAINT NN_ID_HPS_TAB_HSP_EST NOT NULL ,  
    ID_QUARTO  NUMBER (10)  
        CONSTRAINT NN_ID_QRT_TAB_HSP_EST NOT NULL  
)  
  
    PCTFREE 10  
    PCTUSED 40  
    TABLESPACE TBS_NEXUS_DATA_001  
    LOGGING  
    STORAGE (  
        PCTINCREASE 0  
        MINEXTENTS 1  
        MAXEXTENTS UNLIMITED  
        FREELISTS 1  
        FREELIST GROUPS 1  
        BUFFER_POOL DEFAULT  
    )  
;  

```

```
CREATE INDEX SCH_NEXUS.IX_ID_HSP_TAB_HSP_EST ON SCH_NEXUS.TAB_HOSPEDE_ESTADIA
(
  ID_HOSPEDE ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
  PCTINCREASE 0
  MINEXTENTS 1
  MAXEXTENTS UNLIMITED
  FREELISTS 1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT )
LOGGING
;
```

```
GRANT ALL PRIVILEGES
  ON SCH_NEXUS.TAB_HOSPEDE_ESTADIA TO SCH_NEXUS WITH GRANT OPTION
;
```

```
GRANT SELECT
  ON SCH_NEXUS.TAB_HOSPEDE_ESTADIA TO RL_SCH_NEXUS_NEXUS_READ
;
```

```
GRANT INSERT, SELECT, UPDATE
  ON SCH_NEXUS.TAB_HOSPEDE_ESTADIA TO RL_SCH_NEXUS_RECEP_RW
;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE
  ON SCH_NEXUS.TAB_HOSPEDE_ESTADIA TO RL_SCH_NEXUS_ADM_DADOS_RW
;
```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE_ESTADIA
  ADD CONSTRAINT PK_TAB_HSP_EST PRIMARY KEY ( ID_ESTADIA, ID_HOSPEDE, ID_QUARTO ) ;
```

```
CREATE TABLE SCH_NEXUS.TAB_PAGAMENTO
(
  ID_PAGAMENTO NUMBER (10)
```

```

        CONSTRAINT NN_ID_TAB_PGT NOT NULL ,
ID_FATURA    NUMBER (10)

        CONSTRAINT NN_ID_FAT_TAB_PGT NOT NULL ,
VAL_PAGO     NUMBER (10,2)

        CONSTRAINT NN_VAL_PAGO_TAB_PGT NOT NULL ,
MET_PAGAMENTO VARCHAR2 (20 CHAR)

        CONSTRAINT NN_MET_PAG_TAB_PGT NOT NULL ,
DAT_PAGAMENTO DATE

        CONSTRAINT NN_DAT_PAG_TAB_PGT NOT NULL ,
OBS_PAGAMENTO VARCHAR2 (255 CHAR)
)

PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;

CREATE INDEX SCH_NEXUS.IX_ID_FAT_TAB_PGT ON SCH_NEXUS.TAB_PAGAMENTO
(
ID_FATURA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1

```

```

        BUFFER_POOL DEFAULT )
    LOGGING
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS.TAB_PAGAMENTO TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
    ON SCH_NEXUS.TAB_PAGAMENTO TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_PAGAMENTO TO RL_SCH_NEXUS_RECEP_RW
;

GRANT DELETE, INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_PAGAMENTO TO RL_SCH_NEXUS_ADM_DADOS_RW
;

ALTER TABLE SCH_NEXUS.TAB_PAGAMENTO
    ADD CONSTRAINT PK_TAB_PGT PRIMARY KEY ( ID_PAGAMENTO );

CREATE TABLE SCH_NEXUS.TAB_PERIODO_TARIFA
(
    ID_PERIODO NUMBER (10)
        CONSTRAINT NN_ID_PERIODO_TAB_PRD_TAR NOT NULL ,
    NOM_PERIODO VARCHAR2 (50 CHAR)
        CONSTRAINT NN_NOM_PERIODO_TAB_PRD_TAR NOT NULL ,
    DAT_INICIO DATE
        CONSTRAINT NN_DAT_INICIO_TAB_PRD_TAR NOT NULL ,
    DAT_FIM DATE
        CONSTRAINT NN_DAT_FIM_TAB_PRD_TAR NOT NULL
)

PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING

```



```
STORAGE (  
PCTINCREASE 0  
MINEXTENTS 1  
MAXEXTENTS UNLIMITED  
FREELISTS 1  
FREELIST GROUPS 1  
BUFFER_POOL DEFAULT  
)  
;
```

```
GRANT ALL PRIVILEGES  
ON SCH_NEXUS.TAB_PERIODO_TARIFA TO SCH_NEXUS WITH GRANT OPTION  
;  
GRANT SELECT  
ON SCH_NEXUS.TAB_PERIODO_TARIFA TO RL_SCH_NEXUS_NEXUS_READ  
;  
GRANT DELETE, INSERT, SELECT, UPDATE  
ON SCH_NEXUS.TAB_PERIODO_TARIFA TO RL_SCH_NEXUS_ADM_DADOS_RW  
;
```

```
ALTER TABLE SCH_NEXUS.TAB_PERIODO_TARIFA  
ADD CONSTRAINT PK_TAB_PRD_TAR PRIMARY KEY ( ID_PERIODO ) ;
```

```
CREATE TABLE SCH_NEXUS_IDENTITY.TAB_PERMISSAO  
(  
ID_PERMISSAO NUMBER (10)  
CONSTRAINT NN_ID_TAB_PERM NOT NULL ,  
NOM_PERMISSAO VARCHAR2 (50 CHAR)  
CONSTRAINT NN_NOM_TAB_PERM NOT NULL  
)  
PCTFREE 10  
PCTUSED 40  
TABLESPACE TBS_NEXUS_IDENTITY_DATA_001  
LOGGING  
STORAGE (
```

```

PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS_IDENTITY.TAB_PERMISSAO TO SCH_NEXUS_IDENTITY WITH GRANT OPTION
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS_IDENTITY.TAB_PERMISSAO TO RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW
;

GRANT SELECT
    ON SCH_NEXUS_IDENTITY.TAB_PERMISSAO TO RL_SCH_NEXUS_IDENTITY_ID_READ
;

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_PERMISSAO
    ADD CONSTRAINT PK_TAB_PERM PRIMARY KEY ( ID_PERMISSAO );

ALTER TABLE SCH_NEXUS_IDENTITY.TAB_PERMISSAO
    ADD CONSTRAINT AK_NOM_TAB_PERM UNIQUE ( NOM_PERMISSAO );

CREATE TABLE SCH_NEXUS.TAB_QUARTO
(
    ID_QUARTO    NUMBER (10)
        CONSTRAINT NN_ID_TAB_QRT NOT NULL ,
    NUM_QUARTO   VARCHAR2 (10 CHAR)
        CONSTRAINT NN_NUM_TAB_QRT NOT NULL ,
    ID_TIPOQUARTO NUMBER (10)
        CONSTRAINT NN_ID_TP_QRT_TAB_QRT NOT NULL ,
    AND_QUARTO   NUMBER (2)
        CONSTRAINT NN_AND_TAB_QRT NOT NULL ,

```

```

STS_QUARTO  VARCHAR2 (25 CHAR)
    CONSTRAINT NN_STS_TAB_QRT NOT NULL ,
OBS_QUARTO  VARCHAR2 (255 CHAR)
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;
CREATE INDEX SCH_NEXUS.IX_ID_TP_QRT_TAB_QRT ON SCH_NEXUS.TAB_QUARTO
(
ID_TIPOQUARTO ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT )
LOGGING
;
CREATE INDEX SCH_NEXUS.IX_STS_TAB_QRT ON SCH_NEXUS.TAB_QUARTO
(
STS_QUARTO ASC
)

```

```

TABLESPACE TBS_NEXUS_INDEX_001

PCTFREE 10

STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )

LOGGING

;

GRANT ALL PRIVILEGES

    ON SCH_NEXUS.TAB_QUARTO TO SCH_NEXUS WITH GRANT OPTION

;

GRANT SELECT

    ON SCH_NEXUS.TAB_QUARTO TO RL_SCH_NEXUS_NEXUS_READ

;

GRANT UPDATE ( STS_QUARTO )

    ON SCH_NEXUS.TAB_QUARTO TO RL_SCH_NEXUS_LIMPEZA_RW

;

GRANT SELECT

    ON SCH_NEXUS.TAB_QUARTO TO RL_SCH_NEXUS_LIMPEZA_RW

;

GRANT DELETE, INSERT, SELECT, UPDATE

    ON SCH_NEXUS.TAB_QUARTO TO RL_SCH_NEXUS_ADM_DADOS_RW

;

ALTER TABLE SCH_NEXUS.TAB_QUARTO

    ADD CONSTRAINT CK_STS_TAB_QRT

    CHECK (STS_QUARTO IN ('DISPONIVEL', 'OCUPADO', 'LIMPEZA_PENDENTE',
'EM_MANUTENCAO','BLOQUEADO'))

;

```

```
ALTER TABLE SCH_NEXUS.TAB_QUARTO
```

```
ADD CONSTRAINT PK_TAB_QRT PRIMARY KEY ( ID_QUARTO );
```

```
ALTER TABLE SCH_NEXUS.TAB_QUARTO
```

```
ADD CONSTRAINT AK_NUM_TAB_QRT UNIQUE ( NUM_QUARTO );
```

```
CREATE TABLE SCH_NEXUS.TAB_QUARTO_COMODIDADE
```

```
(
```

```
  ID_QUARTO    NUMBER (10)
```

```
    CONSTRAINT NN_ID_QUARTO_TAB_QRT_CMD NOT NULL ,
```

```
  ID_COMODIDADE NUMBER (10)
```

```
    CONSTRAINT NN_ID_COMODIDADE_TAB_QRT_CMD NOT NULL
```

```
)
```

```
  PCTFREE 10
```

```
  PCTUSED 40
```

```
  TABLESPACE TBS_NEXUS_DATA_001
```

```
  LOGGING
```

```
  STORAGE (
```

```
    PCTINCREASE 0
```

```
    MINEXTENTS 1
```

```
    MAXEXTENTS UNLIMITED
```

```
    FREELISTS 1
```

```
    FREELIST GROUPS 1
```

```
    BUFFER_POOL DEFAULT
```

```
)
```

```
;
```

```
GRANT SELECT
```

```
  ON SCH_NEXUS.TAB_QUARTO_COMODIDADE TO RL_SCH_NEXUS_NEXUS_READ
```

```
;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE
```

```
  ON SCH_NEXUS.TAB_QUARTO_COMODIDADE TO RL_SCH_NEXUS_ADM_DADOS_RW
```

```
;
```

```
ALTER TABLE SCH_NEXUS.TAB_QUARTO_COMODIDADE
```

```
ADD CONSTRAINT PK_TAB_QRT_CMD PRIMARY KEY ( ID_QUARTO, ID_COMODIDADE );
```

```

CREATE TABLE SCH_NEXUS.TAB_QUARTO_RESERVADO
(
  ID_RESERVA NUMBER (10)
    CONSTRAINT NN_ID_RES_TAB_QRT_RES NOT NULL ,
  ID_QUARTO  NUMBER (10)
    CONSTRAINT NN_ID_QRT_TAB_QRT_RES NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
  PCTINCREASE 0
  MINEXTENTS 1
  MAXEXTENTS UNLIMITED
  FREELISTS 1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT
)
;

GRANT ALL PRIVILEGES
  ON SCH_NEXUS.TAB_QUARTO_RESERVADO TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
  ON SCH_NEXUS.TAB_QUARTO_RESERVADO TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT INSERT, SELECT, UPDATE
  ON SCH_NEXUS.TAB_QUARTO_RESERVADO TO RL_SCH_NEXUS_RECEP_RW
;

GRANT DELETE, INSERT, SELECT, UPDATE
  ON SCH_NEXUS.TAB_QUARTO_RESERVADO TO RL_SCH_NEXUS_ADM_DADOS_RW
;

ALTER TABLE SCH_NEXUS.TAB_QUARTO_RESERVADO

```

```
ADD CONSTRAINT PK_TAB_QRT_RES PRIMARY KEY ( ID_RESERVA, ID_QUARTO );
```

```
CREATE TABLE SCH_NEXUS.TAB_RESERVA
```

```
(  
  ID_RESERVA      NUMBER (10)  
    CONSTRAINT NN_ID_TAB_RES NOT NULL ,  
  ID_HOSPEDE_PRINCIPAL NUMBER (10)  
    CONSTRAINT NN_ID_HOS_TAB_RES NOT NULL ,  
  DAT_RESERVA      DATE  
    CONSTRAINT NN_DAT_RES_TAB_RES NOT NULL ,  
  DAT_CHECKIN_PREVISTA DATE  
    CONSTRAINT NN_DAT_CKIN_PREV_TAB_RES NOT NULL ,  
  DAT_CHECKOUT_PREVISTA DATE  
    CONSTRAINT NN_DAT_CKOUT_PREV_TAB_RES NOT NULL ,  
  NUM_ADULTOS      NUMBER (2)  
    CONSTRAINT NN_NUM_ADULT_TAB_RES NOT NULL ,  
  NUM_CRIANCAS     NUMBER (2) ,  
  STS_RESERVA      VARCHAR2 (20 CHAR)  
    CONSTRAINT NN_STS_TAB_RES NOT NULL  
)
```

```
PCTFREE 10
```

```
PCTUSED 40
```

```
TABLESPACE TBS_NEXUS_DATA_001
```

```
LOGGING
```

```
STORAGE (
```

```
PCTINCREASE 0
```

```
MINEXTENTS 1
```

```
MAXEXTENTS UNLIMITED
```

```
FREELISTS 1
```

```
FREELIST GROUPS 1
```

```
BUFFER_POOL DEFAULT
```

```
)
```

```
;
```

```
CREATE INDEX SCH_NEXUS.IX_ID_HOSPEDE_TAB_RES ON SCH_NEXUS.TAB_RESERVA
```

```
(  
  ID_HOSPEDE_PRINCIPAL ASC
```

```

)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_DAT_CKIN_TAB_RES ON SCH_NEXUS.TAB_RESERVA
(
    DAT_CHECKIN_PREVISTA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_DAT_CKOUT_TAB_RES ON SCH_NEXUS.TAB_RESERVA
(
    DAT_CHECKOUT_PREVISTA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1

```



```

        MAXEXTENTS UNLIMITED
        FREELISTS 1
        FREELIST GROUPS 1
        BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_STS_TAB_RES ON SCH_NEXUS.TAB_RESERVA
(
    STS_RESERVA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
    PCTINCREASE 0
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
    ON SCH_NEXUS.TAB_RESERVA TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
    ON SCH_NEXUS.TAB_RESERVA TO RL_SCH_NEXUS_NEXUS_READ
;

GRANT INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_RESERVA TO RL_SCH_NEXUS_RECEP_RW
;

GRANT DELETE, INSERT, SELECT, UPDATE
    ON SCH_NEXUS.TAB_RESERVA TO RL_SCH_NEXUS_ADM_DADOS_RW
;

```

```
ALTER TABLE SCH_NEXUS.TAB_RESERVA
ADD CONSTRAINT CK_STS_TAB_RES
CHECK (STS_RESERVA IN ('CONFIRMADA', 'PENDENTE', 'CANCELADA', 'NOSHOW'))
;
```

```
ALTER TABLE SCH_NEXUS.TAB_RESERVA
ADD CONSTRAINT PK_TAB_RES PRIMARY KEY ( ID_RESERVA ) ;
```

```
CREATE TABLE SCH_NEXUS.TAB_SERVICO
(
ID_SERVICO NUMBER (10)
CONSTRAINT NN_ID_TAB_SRV NOT NULL ,
NOM_SERVICO VARCHAR2 (50 CHAR)
CONSTRAINT NN_NOM_TAB_SRV NOT NULL ,
DES_SERVICO VARCHAR2 (500 CHAR)
CONSTRAINT NN_DES_TAB_SRV NOT NULL ,
VAL_SERVICO NUMBER (10,2)
CONSTRAINT NN_VAL_TAB_SRV NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;
```

```
GRANT ALL PRIVILEGES
ON SCH_NEXUS.TAB_SERVICO TO SCH_NEXUS WITH GRANT OPTION
;
```

GRANT SELECT

ON SCH\_NEXUS.TAB\_SERVICO TO RL\_SCH\_NEXUS\_NEXUS\_READ

;

GRANT DELETE, INSERT, SELECT, UPDATE

ON SCH\_NEXUS.TAB\_SERVICO TO RL\_SCH\_NEXUS\_ADM\_DADOS\_RW

;

ALTER TABLE SCH\_NEXUS.TAB\_SERVICO

ADD CONSTRAINT PK\_TAB\_SRV PRIMARY KEY ( ID\_SERVICO );

CREATE TABLE SCH\_NEXUS.TAB\_TARIFA

(

ID\_TARIFA      NUMBER (10)

CONSTRAINT NN\_ID\_TAB\_TAR NOT NULL ,

ID\_TIPO\_QUARTO    NUMBER (10)

CONSTRAINT NN\_TIP\_QRT\_TAB\_TAR NOT NULL ,

ID\_PERIODO\_TARIFA NUMBER (10)

CONSTRAINT NN\_ID\_PER\_TAR\_TAB\_TAR NOT NULL ,

VAL\_DIARIA      NUMBER (10,2)

CONSTRAINT NN\_VAL\_DIA\_TAB\_TAR NOT NULL ,

DIA\_SEMANA      VARCHAR2 (15 CHAR)

CONSTRAINT NN\_DIA\_SEMA\_TAB\_TAR NOT NULL

)

PCTFREE 10

PCTUSED 40

TABLESPACE TBS\_NEXUS\_DATA\_001

LOGGING

STORAGE (

PCTINCREASE 0

MINEXTENTS 1

MAXEXTENTS UNLIMITED

FREELISTS 1

FREELIST GROUPS 1

BUFFER\_POOL DEFAULT

)

;

```
CREATE INDEX SCH_NEXUS.IX_ID_TP_QRT_TAB_TAR ON SCH_NEXUS.TAB_TARIFA
(
  ID_TIPO_QUARTO ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
  PCTINCREASE 0
  MINEXTENTS 1
  MAXEXTENTS UNLIMITED
  FREELISTS 1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT )
LOGGING
;

CREATE INDEX SCH_NEXUS.IX_ID_PER_TAR_TAB_TAR ON SCH_NEXUS.TAB_TARIFA
(
  ID_PERIODO_TARIFA ASC
)
TABLESPACE TBS_NEXUS_INDEX_001
PCTFREE 10
STORAGE (
  PCTINCREASE 0
  MINEXTENTS 1
  MAXEXTENTS UNLIMITED
  FREELISTS 1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT )
LOGGING
;

GRANT ALL PRIVILEGES
  ON SCH_NEXUS.TAB_TARIFA TO SCH_NEXUS WITH GRANT OPTION
;

GRANT SELECT
```

```
ON SCH_NEXUS.TAB_TARIFA TO RL_SCH_NEXUS_NEXUS_READ
;
GRANT DELETE, INSERT, SELECT, UPDATE
ON SCH_NEXUS.TAB_TARIFA TO RL_SCH_NEXUS_ADM_DADOS_RW
;
```

```
ALTER TABLE SCH_NEXUS.TAB_TARIFA
ADD CONSTRAINT PK_TAB_TAR PRIMARY KEY ( ID_TARIFA );
```

```
CREATE TABLE SCH_NEXUS.TAB_TIPO_QUARTO
(
ID_TIPO  NUMBER (10)
CONSTRAINT NN_ID_TAB_TP_QRT NOT NULL ,
NOM_TIPO  VARCHAR2 (50 CHAR)
CONSTRAINT NN_NOM_TIPO_TAB_TP_QRT NOT NULL ,
CAP_MAXIMA  NUMBER (2)
CONSTRAINT NN_CAP_MAX_TAB_TP_QRT NOT NULL ,
DESC_TIPO  VARCHAR2 (500 CHAR)
CONSTRAINT NN_DESC_TIPO_TAB_TP_QRT NOT NULL
)
PCTFREE 10
PCTUSED 40
TABLESPACE TBS_NEXUS_DATA_001
LOGGING
STORAGE (
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENTS UNLIMITED
FREELISTS 1
FREELIST GROUPS 1
BUFFER_POOL DEFAULT
)
;
```

```
GRANT ALL PRIVILEGES
```

ON SCH\_NEXUS.TAB\_TIPO\_QUARTO TO SCH\_NEXUS WITH GRANT OPTION

;

GRANT SELECT

ON SCH\_NEXUS.TAB\_TIPO\_QUARTO TO RL\_SCH\_NEXUS\_NEXUS\_READ

;

GRANT DELETE, INSERT, SELECT, UPDATE

ON SCH\_NEXUS.TAB\_TIPO\_QUARTO TO RL\_SCH\_NEXUS\_ADM\_DADOS\_RW

;

ALTER TABLE SCH\_NEXUS.TAB\_TIPO\_QUARTO

ADD CONSTRAINT PK\_TAB\_TP\_QRT PRIMARY KEY ( ID\_TIPO );

ALTER TABLE SCH\_NEXUS.TAB\_TIPO\_QUARTO

ADD CONSTRAINT AK\_NOM\_TIPO\_TAB\_TP\_QRT UNIQUE ( NOM\_TIPO );

ALTER TABLE SCH\_NEXUS\_IDENTITY.TAB\_CARGO\_PERMISSAO

ADD CONSTRAINT FK\_CARGO\_TAB\_CRG\_PERM FOREIGN KEY

(

ID\_CARGO

)

REFERENCES SCH\_NEXUS\_IDENTITY.TAB\_CARGO

(

ID\_CARGO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.TAB\_FUNCIONARIO

ADD CONSTRAINT FK\_CARGO\_TAB\_FUNC FOREIGN KEY

(

ID\_CARGO

)

REFERENCES SCH\_NEXUS\_IDENTITY.TAB\_CARGO

(

ID\_CARGO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_QUARTO\_COMODIDADE

ADD CONSTRAINT FK\_COMODIDADE FOREIGN KEY

(

ID\_COMODIDADE

)

REFERENCES SCH\_NEXUS.TAB\_COMODIDADE

(

ID\_COMODIDADE

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_FATURA

ADD CONSTRAINT FK\_EST\_TAB\_FAT FOREIGN KEY

(

ID\_ESTADIA

)

REFERENCES SCH\_NEXUS.TAB\_ESTADIA

(

ID\_ESTADIA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_AVALIACAO

ADD CONSTRAINT FK\_ESTADIA\_TAB\_AVL FOREIGN KEY

(

ID\_ESTADIA

)

REFERENCES SCH\_NEXUS.TAB\_ESTADIA

(

ID\_ESTADIA

)

NOT DEFERRABLE

;

```
ALTER TABLE SCH_NEXUS.TAB_CONSUMO_SERVICO
ADD CONSTRAINT FK_ESTADIA_TAB_CNS_SRV FOREIGN KEY
(
    ID_ESTADIA
)
REFERENCES SCH_NEXUS.TAB_ESTADIA
(
    ID_ESTADIA
)
NOT DEFERRABLE
;
```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE_ESTADIA
ADD CONSTRAINT FK_ESTADIA_TAB_HSP_EST FOREIGN KEY
(
    ID_ESTADIA
)
REFERENCES SCH_NEXUS.TAB_ESTADIA
(
    ID_ESTADIA
)
NOT DEFERRABLE
;
```

```
ALTER TABLE SCH_NEXUS.TAB_PAGAMENTO
ADD CONSTRAINT FK_FAT_TAB_PGT FOREIGN KEY
(
    ID_FATURA
)
REFERENCES SCH_NEXUS.TAB_FATURA
(
    ID_FATURA
)
```



NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_AVALIACAO

ADD CONSTRAINT FK\_HOSPEDE\_TAB\_AVL FOREIGN KEY

(

ID\_HOSPEDE

)

REFERENCES SCH\_NEXUS.TAB\_HOSPEDE

(

ID\_HOSPEDE

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_HOSPEDE\_ESTADIA

ADD CONSTRAINT FK\_HOSPEDE\_TAB\_HSP\_EST FOREIGN KEY

(

ID\_HOSPEDE

)

REFERENCES SCH\_NEXUS.TAB\_HOSPEDE

(

ID\_HOSPEDE

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_RESERVA

ADD CONSTRAINT FK\_HOSPEDE\_TAB\_RES FOREIGN KEY

(

ID\_HOSPEDE\_PRINCIPAL

)

REFERENCES SCH\_NEXUS.TAB\_HOSPEDE

(

ID\_HOSPEDE

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_TARIFA

ADD CONSTRAINT FK\_PER\_TAR\_TAB\_TAR FOREIGN KEY

(

ID\_PERIODO\_TARIFA

)

REFERENCES SCH\_NEXUS.TAB\_PERIODO\_TARIFA

(

ID\_PERIODO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS\_IDENTITY.TAB\_CARGO\_PERMISSAO

ADD CONSTRAINT FK\_PERMISSAO\_TAB\_CRG\_PERM FOREIGN KEY

(

ID\_PERMISSAO

)

REFERENCES SCH\_NEXUS\_IDENTITY.TAB\_PERMISSAO

(

ID\_PERMISSAO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_QUARTO\_COMODIDADE

ADD CONSTRAINT FK\_QUARTO FOREIGN KEY

(

ID\_QUARTO

)

REFERENCES SCH\_NEXUS.TAB\_QUARTO

(

ID\_QUARTO

)

NOT DEFERRABLE

;

```
ALTER TABLE SCH_NEXUS.TAB_CONSUMO_SERVICO
ADD CONSTRAINT FK_QUARTO_TAB_CNS_SRV FOREIGN KEY
(
    ID_QUARTO
)
REFERENCES SCH_NEXUS.TAB_QUARTO
(
    ID_QUARTO
)
NOT DEFERRABLE
;
```

```
ALTER TABLE SCH_NEXUS.TAB_HOSPEDE_ESTADIA
ADD CONSTRAINT FK_QUARTO_TAB_HSP_EST FOREIGN KEY
(
    ID_QUARTO
)
REFERENCES SCH_NEXUS.TAB_QUARTO
(
    ID_QUARTO
)
NOT DEFERRABLE
;
```

```
ALTER TABLE SCH_NEXUS.TAB_QUARTO_RESERVADO
ADD CONSTRAINT FK_QUARTO_TAB_QRT_RES FOREIGN KEY
(
    ID_QUARTO
)
REFERENCES SCH_NEXUS.TAB_QUARTO
(
    ID_QUARTO
)
```

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_ESTADIA

ADD CONSTRAINT FK\_RES\_TAB\_EST FOREIGN KEY

(

ID\_RESERVA

)

REFERENCES SCH\_NEXUS.TAB\_RESERVA

(

ID\_RESERVA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_QUARTO\_RESERVADO

ADD CONSTRAINT FK\_RESERVA\_TAB\_QRT\_RES FOREIGN KEY

(

ID\_RESERVA

)

REFERENCES SCH\_NEXUS.TAB\_RESERVA

(

ID\_RESERVA

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_CONSUMO\_SERVICO

ADD CONSTRAINT FK\_SERVICO\_TAB\_CNS\_SRV FOREIGN KEY

(

ID\_SERVICO

)

REFERENCES SCH\_NEXUS.TAB\_SERVICO

(

ID\_SERVICO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_QUARTO

ADD CONSTRAINT FK\_TP\_QRT\_TAB\_QRT FOREIGN KEY

(

ID\_TIPOQUARTO

)

REFERENCES SCH\_NEXUS.TAB\_TIPO\_QUARTO

(

ID\_TIPO

)

NOT DEFERRABLE

;

ALTER TABLE SCH\_NEXUS.TAB\_TARIFA

ADD CONSTRAINT FK\_TP\_QRT\_TAB\_TAR FOREIGN KEY

(

ID\_TIPO\_QUARTO

)

REFERENCES SCH\_NEXUS.TAB\_TIPO\_QUARTO

(

ID\_TIPO

)

NOT DEFERRABLE

;

CREATE OR REPLACE VIEW VW\_CHECKOUT\_DIA ( ID\_ESTADIA, NOME\_HOSPEDE, NUM\_QUARTO,  
DAT\_CHECKOUT\_PREVISTA ) AS

SELECT

e.ID\_ESTADIA,

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

q.NUM\_QUARTO,

r.DAT\_CHECKOUT\_PREVISTA

FROM

SCH\_NEXUS.TAB\_ESTADIA e

JOIN

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

JOIN

SCH\_NEXUS.TAB\_HOSPEDE h ON r.ID\_HOSPEDE\_PRINCIPAL = h.ID\_HOSPEDE

JOIN

SCH\_NEXUS.TAB\_HOSPEDE\_ESTADIA he ON e.ID\_ESTADIA = he.ID\_ESTADIA

JOIN

SCH\_NEXUS.TAB\_QUARTO q ON he.ID\_QUARTO = q.ID\_QUARTO

WHERE

e.STS\_ESTADIA = 'HOSPEDADO'

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

;

GRANT SELECT

ON VW\_CHECKOUT\_DIA TO RL\_SCH\_NEXUS\_CONSULTA\_VW

;

CREATE OR REPLACE VIEW VW\_FUNCIONARIO ( 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.TAB\_FUNCIONARIO f

JOIN

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

;

CREATE OR REPLACE VIEW VW\_QUARTO\_OCUPADO ( NUM\_QUARTO, NOM\_TIPO, AND\_QUARTO, STS\_QUARTO, HOSPEDE\_PRINCIPAL, DAT\_CHECKOUT\_PREVISTA ) AS

SELECT

q.NUM\_QUARTO,

```

    tq.NOM_TIPO,
    q.AND_QUARTO,
    q.STS_QUARTO,
    h.PNM_HOSPEDE || ' ' || h.UNM_HOSPEDE AS HOSPEDE_PRINCIPAL,
    r.DAT_CHECKOUT_PREVISTA
FROM
    SCH_NEXUS.TAB_QUARTO q
INNER JOIN
    SCH_NEXUS.TAB_TIPO_QUARTO tq ON q.ID_TIPOQUARTO = tq.ID_TIPO
LEFT JOIN
    SCH_NEXUS.TAB_HOSPEDE_ESTADIA he ON q.ID_QUARTO = he.ID_QUARTO
LEFT JOIN
    SCH_NEXUS.TAB_ESTADIA e ON he.ID_ESTADIA = e.ID_ESTADIA AND e.STS_ESTADIA = 'HOSPEDADO'
LEFT JOIN
    SCH_NEXUS.TAB_RESERVA r ON e.ID_RESERVA = r.ID_RESERVA
LEFT JOIN
    SCH_NEXUS.TAB_HOSPEDE h ON r.ID_HOSPEDE_PRINCIPAL = h.ID_HOSPEDE
ORDER BY
    q.NUM_QUARTO
;

```

```

GRANT SELECT
ON VW_QUARTO_OCUPADO TO RL_SCH_NEXUS_CONSULTA_VW
;

```

```

CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_AVL
START WITH 1
    NOCACHE
    ORDER ;

```

```

CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_AVL
BEFORE INSERT ON SCH_NEXUS.TAB_AVALIACAO
FOR EACH ROW
WHEN (NEW.ID_AVALIACAO IS NULL)

```

```

BEGIN
    :NEW.ID_AVALIACAO := SCH_NEXUS.SEQ_ID_TAB_AVL.NEXTVAL;
END;
/

CREATE SEQUENCE SCH_NEXUS_IDENTITY.SEQ_ID_TAB_CRG
START WITH 1
    NOCACHE
    ORDER ;

CREATE OR REPLACE TRIGGER SCH_NEXUS_IDENTITY.TRG_ID_TAB_CRG
BEFORE INSERT ON SCH_NEXUS_IDENTITY.TAB_CARGO
FOR EACH ROW
WHEN (NEW.ID_CARGO IS NULL)
BEGIN
    :NEW.ID_CARGO := SCH_NEXUS_IDENTITY.SEQ_ID_TAB_CRG.NEXTVAL;
END;
/

CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_CMD
START WITH 1
    NOCACHE
    ORDER ;

CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_CMD
BEFORE INSERT ON SCH_NEXUS.TAB_COMODIDADE
FOR EACH ROW
WHEN (NEW.ID_COMODIDADE IS NULL)
BEGIN
    :NEW.ID_COMODIDADE := SCH_NEXUS.SEQ_ID_TAB_CMD.NEXTVAL;
END;
/

CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_CNS_SRV
START WITH 1
    NOCACHE

```



ORDER ;

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_CNS_SRV
BEFORE INSERT ON SCH_NEXUS.TAB_CONSUMO_SERVICO
FOR EACH ROW
WHEN (NEW.ID_CONSUMO IS NULL)
BEGIN
    :NEW.ID_CONSUMO := SCH_NEXUS.SEQ_ID_TAB_CNS_SRV.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_EST
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_EST
BEFORE INSERT ON SCH_NEXUS.TAB_ESTADIA
FOR EACH ROW
WHEN (NEW.ID_ESTADIA IS NULL)
BEGIN
    :NEW.ID_ESTADIA := SCH_NEXUS.SEQ_ID_TAB_EST.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_FAT
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_FAT
BEFORE INSERT ON SCH_NEXUS.TAB_FATURA
FOR EACH ROW
WHEN (NEW.ID_FATURA IS NULL)
BEGIN
    :NEW.ID_FATURA := SCH_NEXUS.SEQ_ID_TAB_FAT.NEXTVAL;
```

END;

/

CREATE SEQUENCE SCH\_NEXUS\_IDENTITY.SEQ\_ID\_TAB\_FUNC

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS\_IDENTITY.TRG\_ID\_TAB\_FUNC

BEFORE INSERT ON SCH\_NEXUS\_IDENTITY.TAB\_FUNCIONARIO

FOR EACH ROW

WHEN (NEW.ID\_FUNCIONARIO IS NULL)

BEGIN

:NEW.ID\_FUNCIONARIO := SCH\_NEXUS\_IDENTITY.SEQ\_ID\_TAB\_FUNC.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.SEQ\_ID\_TAB\_HSP

START WITH 1

NOCACHE

ORDER ;

CREATE OR REPLACE TRIGGER SCH\_NEXUS.TRG\_ID\_TAB\_HSP

BEFORE INSERT ON SCH\_NEXUS.TAB\_HOSPEDE

FOR EACH ROW

WHEN (NEW.ID\_HOSPEDE IS NULL)

BEGIN

:NEW.ID\_HOSPEDE := SCH\_NEXUS.SEQ\_ID\_TAB\_HSP.NEXTVAL;

END;

/

CREATE SEQUENCE SCH\_NEXUS.SEQ\_ID\_TAB\_PGT

START WITH 1

NOCACHE

ORDER ;

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_PGT
BEFORE INSERT ON SCH_NEXUS.TAB_PAGAMENTO
FOR EACH ROW
WHEN (NEW.ID_PAGAMENTO IS NULL)
BEGIN
    :NEW.ID_PAGAMENTO := SCH_NEXUS.SEQ_ID_TAB_PGT.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_PRD_TAR
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_PRD_TAR
BEFORE INSERT ON SCH_NEXUS.TAB_PERIODO_TARIFA
FOR EACH ROW
WHEN (NEW.ID_PERIODO IS NULL)
BEGIN
    :NEW.ID_PERIODO := SCH_NEXUS.SEQ_ID_TAB_PRD_TAR.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS_IDENTITY.SEQ_ID_TAB_PERM
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS_IDENTITY.TRG_ID_TAB_PERM
BEFORE INSERT ON SCH_NEXUS_IDENTITY.TAB_PERMISSAO
FOR EACH ROW
WHEN (NEW.ID_PERMISSAO IS NULL)
BEGIN
    :NEW.ID_PERMISSAO := SCH_NEXUS_IDENTITY.SEQ_ID_TAB_PERM.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_QRT
START WITH 1
  NOCACHE
  ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_QRT
BEFORE INSERT ON SCH_NEXUS.TAB_QUARTO
FOR EACH ROW
WHEN (NEW.ID_QUARTO IS NULL)
BEGIN
  :NEW.ID_QUARTO := SCH_NEXUS.SEQ_ID_TAB_QRT.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_RES
START WITH 1
  NOCACHE
  ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_RES
BEFORE INSERT ON SCH_NEXUS.TAB_RESERVA
FOR EACH ROW
WHEN (NEW.ID_RESERVA IS NULL)
BEGIN
  :NEW.ID_RESERVA := SCH_NEXUS.SEQ_ID_TAB_RES.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_SRV
START WITH 1
  NOCACHE
  ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_SRV
BEFORE INSERT ON SCH_NEXUS.TAB_SERVICO
```

```
FOR EACH ROW
WHEN (NEW.ID_SERVICO IS NULL)
BEGIN
    :NEW.ID_SERVICO := SCH_NEXUS.SEQ_ID_TAB_SRV.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_TAR
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_TAR
BEFORE INSERT ON SCH_NEXUS.TAB_TARIFA
FOR EACH ROW
WHEN (NEW.ID_TARIFA IS NULL)
BEGIN
    :NEW.ID_TARIFA := SCH_NEXUS.SEQ_ID_TAB_TAR.NEXTVAL;
END;
/
```

```
CREATE SEQUENCE SCH_NEXUS.SEQ_ID_TAB_TP_QRT
START WITH 1
    NOCACHE
    ORDER ;
```

```
CREATE OR REPLACE TRIGGER SCH_NEXUS.TRG_ID_TAB_TP_QRT
BEFORE INSERT ON SCH_NEXUS.TAB_TIPO_QUARTO
FOR EACH ROW
WHEN (NEW.ID_TIPO IS NULL)
BEGIN
    :NEW.ID_TIPO := SCH_NEXUS.SEQ_ID_TAB_TP_QRT.NEXTVAL;
END;
/
```

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

--

-- CREATE TABLE	20	
-- CREATE INDEX	20	
-- ALTER TABLE	55	
-- CREATE VIEW	3	
-- ALTER VIEW	0	
-- CREATE PACKAGE	0	
-- CREATE PACKAGE BODY	0	
-- CREATE PROCEDURE	0	
-- CREATE FUNCTION	0	
-- CREATE TRIGGER	16	
-- 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	11	
-- CREATE ROLLBACK SEGMENT	0	
-- CREATE SEQUENCE	16	
-- CREATE MATERIALIZED VIEW	0	
-- CREATE MATERIALIZED VIEW LOG	0	
-- CREATE SYNONYM	0	
-- CREATE TABLESPACE	4	
-- CREATE USER	6	
--		
-- 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
```