

Projeto:	Sistema de gestão de Hotéis/Pousadas	Versão:	1.0.3
Autores:	João Pedro de Castro Laranjeira Rocha e Daniel do Carmo Nascimento	Última Alteração:	19/10/2025

- 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.

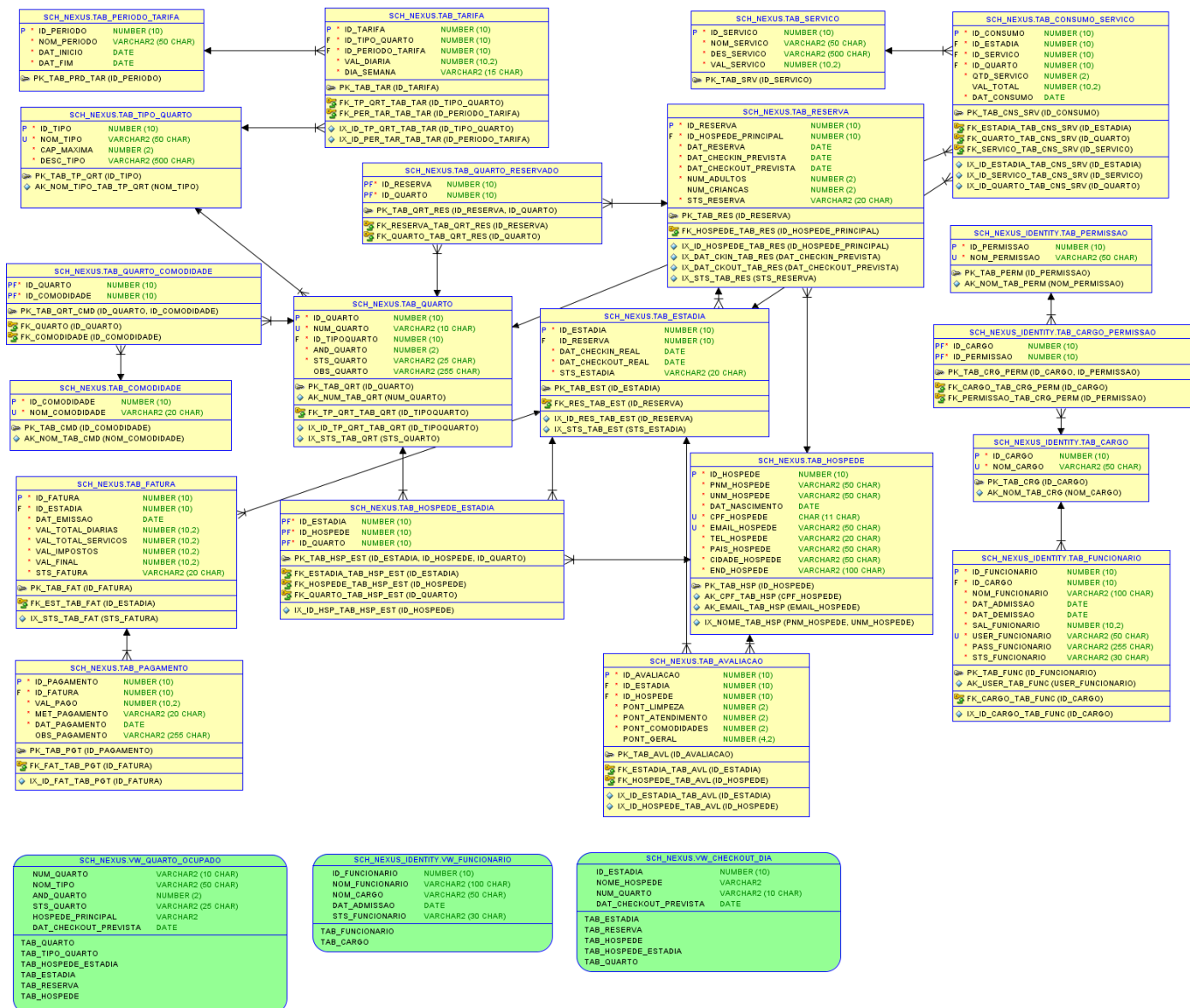
Funcionários e Controle de Acesso

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.

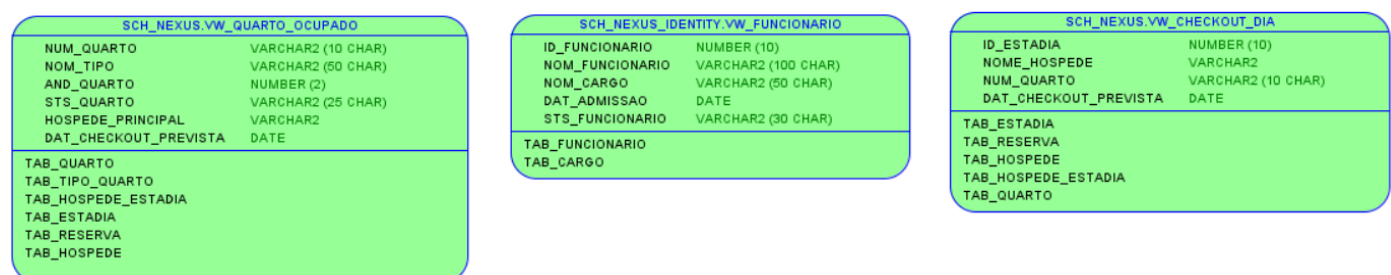
Todo funcionário ocupa um cargo (ex: 'Recepcionista', 'Gerente', 'Camareira'). Cada cargo, por sua vez, está associado a um conjunto de permissões que definem as ações que o usuário pode realizar no sistema. Um cargo pode ter várias permissões, e uma permissão pode ser atribuída a vários cargos.

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_CONSULTA_VW	SELECT	SCH_NEXUS	VW_CHECKOUT_DIA, VW_QUARTO_OCUPADO
RL_SCH_NEXUS_IDENTITY_ID_FUNC_RW	SELECT, INSERT, UPDATE	SCH_NEXUS_IDENTITY	TAB_FUNCIONARIO, TAB_CARGO, TAB_PERMISSAO, TAB_CARGO_PERMISSAO
RL_SCH_NEXUS_IDENTITY_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

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

13. PLANO DE RECUPERAÇÃO DE DESASTRES

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: 2025-10-19 17:36:21 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_USUARIO_PADRAO;

--- 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_USUARIO_PADRAO 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;
```

```
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_NE
XUS_IDENTITY_ID_READ TO GP_RECEPCIONISTA

;

CREATE USER DANIEL_RECEPCIONISTA

IDENTIFIED BY daniel123

PROFILE PRO_USUARIO_PADRAO

ACCOUNT UNLOCK

;

GRANT GP_RECEPCIONISTA TO DANIEL_RECEPCIONISTA

;

CREATE USER DAVI_CLIENTE

IDENTIFIED BY joao1234

PROFILE PRO_USUARIO_PADRAO

ACCOUNT UNLOCK

;

GRANT GP_CLIENTE TO DAVI_CLIENTE

;

CREATE USER JOAO_SERV_GERAIS

IDENTIFIED BY joao123

PROFILE PRO_USUARIO_PADRAO

ACCOUNT UNLOCK

;

GRANT GP_LIMPEZA TO JOAO_SERV_GERAIS

;

CREATE USER MARIA_GERENTE

IDENTIFIED BY maria_gerente

PROFILE PRO_USUARIO_PADRAO

ACCOUNT UNLOCK

;

GRANT GP_GERENTE TO MARIA_GERENTE

;

CREATE USER SCH_NEXUS

IDENTIFIED BY sch_nexus_123

PASSWORD EXPIRE

ACCOUNT LOCK

;

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.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	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	0
-- CREATE SEQUENCE	16	
-- CREATE MATERIALIZED VIEW	0	
-- CREATE MATERIALIZED VIEW LOG	0	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	