

EXEMPLO DE CRIAÇÃO DE OBJETOS:

Atenção às cláusulas destacadas.

1) CRIAÇÃO DA TABELA NORMAL

```
CREATE TABLE SIFH.CPA_ANALISE_STATUS
( ANST_SQ          NUMBER          NOT NULL
, ANST_DS          VARCHAR2(35)    NOT NULL
, ANST_LOGIN       VARCHAR2(35)    NOT NULL
, ANST_DT_INICIO   DATE            NOT NULL
, ANST_DT_TERMINO  DATE)
LOGGING --> padrão
NOCOMPRESS --> padrão
NOCACHE --> padrão
NOPARALLEL --> padrão
MONITORING; --> padrão;
```

2) CRIAÇÃO DE ATRIBUTOS COM TIPO CLOB OU BLOB

A criação do atributo com datatype LOB deve ser sempre direcionada ao tablespace destinado a manter este tipo de segmento

Exemplos:

a) Durante a criação de uma tabela:

```
CREATE TABLE ACESSO.USU_MODULO
(MODU_NR          NUMBER NOT NULL,
- DEMAIS ATRIBUTOS
- DEMAIS ATRIBUTOS
- DEMAIS ATRIBUTOS
MODU_IMAGEM       BLOB) --> nome do atributo LOB
LOB (MODU_IMAGEM) STORE AS MODU_IMAGEM --> nome do objeto no tablespace LOB
( TABLESPACE SI3_LOB --> tablespace para atributo LOB
  ENABLE STORAGE IN ROW
  CHUNK 8192
  PCTVERSION 10
);
```

b) Adição de campo LOB em tabela existente:

```
ALTER TABLE ACESSO.USU_MODULO ADD (MODU_IMAGEM BLOB)
LOB (MODU_IMAGEM) STORE AS MODU_IMAGEM (TABLESPACE SI3_LOB);
```

3) CRIAÇÃO DE PRIMARY KEY

```
ALTER TABLE SIFH.CPA_ANALISE_STATUS
ADD( CONSTRAINT ANST_PK PRIMARY KEY( ANST_SQ ) USING INDEX TABLESPACE SI3_INDEX
COMPUTE STATISTICS);
```

4) CRIAÇÃO DE UNIQUE KEY

```
ALTER TABLE SIFH.CPA_ANALISE_STATUS
ADD( CONSTRAINT ANST_UK UNIQUE( ANST_DS ) USING INDEX TABLESPACE SI3_INDEX COMPUTE
STATISTICS) ;
```

5) CRIAÇÃO DE ÍNDICE

```
CREATE INDEX SIFH.ANST_LOGIN_FK_I ON SIFH.CPA_ANALISE_STATUS( ANST_LOGIN )
TABLESPACE SI3_INDEX COMPUTE STATISTICS;
```

6) CRIAÇÃO DE ÍNDICE PARA CHAVES ESTRANGEIRAS (FK) (Ver item 8)

Após a criação da FK é necessária a criação do índice correspondente na tabela filha (a que mantém a FK) desde a versão 9i, devido à mudanças no sistema de gerenciamento de locks.

Exemplo:

ADM_ADMISSAO mantém constraint com a PAC_PACIENTE pela coluna ADM_PACI_ID

Neste caso, deve ser criado um índice na coluna ADM_PACI_ID, o qual seria utilizado na seguinte query:

```
select adm_inst_cd, adm_ano, adm_nr, adm_dt_ingresso
from   pac_paciente, adm_admissao
where  adm_paci_id = paci_id
and    paci_id = 6844
```

Se o índice na coluna ADM_PACI_ID não existisse, haveria lentidão generalizada, provocada pela concorrência das sessões.

7) PRIVILÉGIOS E SINÔNIMOS.

Após a criação de objetos que façam uso de sinônimos e privilégios de acesso, utilize a procedure PRC_GRANTS_SI3 para criá-los

Esta procedure cria o sinônimo publico (com o mesmo nome do objeto) e concede privilégios da seguinte forma:

SELECT - em tabelas e views para os usuários ADM_RO e REPORTING

SELECT, INSERT, UPDATE, DELETE – em tabelas e views para os usuários: ADM, ADM_RW, SI3_PATCH e todos os owners do SI3.

EXECUTE – em procedures, functions, packages e types para os usuários: ADM, ADM_RW, SI3_PATCH e todos os owners do SI3.

Os parâmetros da procedure são owner e nome do objeto

Exemplo:

```
EXECUTE PRC_GRANTS_SI3( 'SIFH', 'CPA_ANALISE_STATUS' );
```

8) CRIAÇÃO DE CÓPIAS DE SEGURANÇA DE TABELAS

Muitas vezes há a necessidade de manter-se uma cópia de segurança de tabelas envolvidas em uma operação de risco.

Tais objetos devem ser criados no schema **COP_SEG**, criado especificamente com este objetivo: não poluir os schemas do SI3 com tabelas de cópia.

Sugerimos que a tabela seja criada com o formato INICIAIS DO ANALISTA_NOME DA TABELA ou NUMERO DA OS_NOME DA TABELA.

A data de criação não é necessária pois pode ser verificada pela equipe de bancos de dados, caso necessário.

9) CRIAÇÃO DE JOBS

Visando facilitar a inclusão da criação de jobs nos patches do SI3 foi criada procedure PRC_CRIA_JOB

NOTA: Há um template para o uso da procedure, disponível no Source Safe em :
\$/SI3/NEW/DB_OBJECTS/JOBS/TEMPLATE_JOB.job

IMPORTANTE: No leia-me do patch deve constar que está sendo criado um job Oracle, a procedure que vai ser executada e o seu intervalo.

PARAMETROS DA PROCEDURE:

P_USER in varchar -- Informar o owner do job/objeto a ser executado. Caso seja algum bloco plsql, informar

o owner do objeto ou módulo principal.

P_NEXT_DATE in date - Data da primeira execução do job

Exemplos:

Iniciar execução na data de criação do job, às 20:00

TRUNC(SYSDATE)+20/24

Iniciar execução no dia seguinte à criação do job, às 06:00

TRUNC(SYSDATE+1)+6/24

P_INTERVAL in varchar2 - Intervalo de execução do job

Exemplos:

A cada meia hora: SYSDATE+30/1440

Toda segunda-feira às 15:00: NEXT_DAY(TRUNC(SYSDATE), 'MONDAY') + 15/24

Diariamente às 08:00: TRUNC(SYSDATE + 1) + 8/24

Primeiro dia de cada mês às 08:00: TRUNC(LAST_DAY(SYSDATE) + 1) + 8/24

P_WHAT in varchar2 - Código a ser executado

NOTA: Favor seguir o template para documentação do job.

IMPORTANTE:

No log do job informe o nome do job entre aspas duplas:

Exemplo:

```
INSERT INTO JOB_LOG (log_job, log_inicio, log_nome, log_tipo)
VALUES (v_job, SYSDATE, 'PACIENTE.PRC_EMAIL_NOTIFIC_CIRURGIA', 'SI3');
```

EXEMPLO DE CRIAÇÃO DO JOB:

OWNER DO OBJETO: PACIENTE

OBJETO: PRC_EMAIL_NOTIFIC_CIRURGIA

INTERVALO: Diariamente AS 20:00

PRIMEIRA EXECUÇÃO: No dia da criação do job, às 20:00

SCRIPT: [\\$/SI3/NEW/DB_OBJECTS/JOB/PACIENTE_PRC_EMAIL_NOTIFIC_CIRURGIA.job Version 2](#)

```
declare
begin
  prc_cria_job (
    'PACIENTE'
    , TRUNC(SYSDATE)+20/24
    , 'TRUNC(SYSDATE+1)+20/24'
    , '/*PACIENTE.PRC_EMAIL_NOTIFIC_CIRURGIA (20:00)
```

OBJETIVO: Job para enviar email para notificação de cirurgias agendadas

AUTOR: ANIBAL CASTILHO LOPES

DATA: 05/12/2012 17:56

OS: 137400

OBSERVAÇÕES:

```
-----
--- PROGRAMAR PARA EXECUTAR 2X AO DIA COM A
--- MESMA DIFERENÇA DE HORAS QUE O PASSADO
--- NO PARAMETRO DA PROCEDURE
--- SUGESTÃO : A SEGUNDA EXECUÇÃO APÓS 3 HORAS
```

```

-----
*/
DECLARE
  v_job NUMBER;
  v_erro VARCHAR2(4000);
BEGIN
  SELECT sq_job.NEXTVAL
  INTO   v_job
  FROM   DUAL;
  --
  INSERT INTO JOB_LOG (log_job,log_inicio,log_nome,log_tipo)
  VALUES (v_job, SYSDATE, 'PACIENTE.PRC_EMAIL_NOTIFIC_CIRURGIA','SI3');
  COMMIT;
  ---
  -----
  --- INICIO
  -----

  PRC_EMAIL_NOTIFIC_CIRURGIA( P_HORAS => 3 ) ;

  ---
  -----
  --- FIM
  -----
  --
  UPDATE JOB_LOG
  SET    log_fim = SYSDATE
  WHERE  log_job = v_job;
  COMMIT;
  --
  EXCEPTION WHEN OTHERS THEN
  v_erro:=SQLERRM;
  UPDATE JOB_LOG
  SET    log_erro = v_erro
  WHERE  log_job = v_job;
  COMMIT;
END;');
END;
/
COMMIT;

```

10) CRIAÇÃO DE USUÁRIOS

O script de criação de novos usuários deve fazer parte de um patch do si3.

```

CREATE USER <nome do usuário>
  IDENTIFIED BY <senha do usuário>
  DEFAULT TABLESPACE SI3_DATA
  TEMPORARY TABLESPACE SI3_TEMP
  PROFILE <tipo de profile>
  ACCOUNT UNLOCK
/
GRANT CONNECT TO <nome do usuário>
/
ALTER USER <nome do usuário> DEFAULT ROLE ALL
/
grant <tipo de grant> on <owner>.<nome do objeto> to <nome do usuário>;

```

grant <tipo de grant> on <owner>.<nome do objeto> to <nome do usuário>;

...

Nome do usuário – Quais são as recomendações ? (<nome usuário>_SI3 quando utilizar ?)

Senha do usuário – pode ser posta a mesma do nome do usuário porém é necessário documentar no patchinfo que a mesma precisa ser trocada conforme a política de segurança.

Tipos de perfis do si3:

SI3: Agrupa as contas que fazem parte do SI³ e mantém seus objetos

DBA: Agrupa as contas dos dbas

ORACLE: Agrupa as contas padrão da instalação Oracle

MANUT: Agrupa as contas necessárias para manutenção do banco de dados e do SI³

APLICACAO: Agrupa as contas de outras aplicações que venham a ser instaladas no mesmo banco de dados do SI³ (embora recomenda-se que o banco de dados seja dedicado ao SI³)

INTEGRACAO: Agrupa as contas que irão manter alguma integração com o SI³. A conta 'infosaude_si3', por exemplo, é criada com este profile pois tem o objetivo de integrar o sistema Infosaude da Salux com o SI³

DBLINK_IN: Agrupa as contas utilizadas por conexões via dblink vindas de outros bancos de dados.

Exemplo - trecho do patch 161141

```
CREATE USER ERITEL_SI3_ACESSO
IDENTIFIED BY ERITEL_SI3_ACESSO
DEFAULT TABLESPACE SI3_DATA
TEMPORARY TABLESPACE SI3_TEMP
PROFILE DBLINK_IN
ACCOUNT UNLOCK
/
```

```
GRANT CONNECT TO ERITEL_SI3_ACESSO
/
```

```
ALTER USER ERITEL_SI3_ACESSO DEFAULT ROLE ALL
/
```

```
grant execute on PACIENTE.pkg_alocacao to ERITEL_SI3;
```