


	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

Índice

1. Introducción	2
2. Objetivos y alcance.....	2
3. Normas Generales.	3
3.1. Uso de mayúsculas y minúsculas.....	3
3.2. Líneas.....	3
4. Formato del Código.	4
4.1. Alineación e Identación.	4
4.1.1. Bloque anónimo.	4
4.1.2. Condiciones.	4
4.1.3. Bucles.....	4
4.1.4. Sentencia SELECT.	5
4.1.5. Sentencias Insert/Delete/Update.	5
4.1.6. Expresiones largas.....	5
5. Comentarios.	7
5.1. Consideraciones Generales.....	7
5.2. Dónde utilizarlos.....	7
5.3. Estilo de los comentarios	7
6. Documentación interna	9
6.1. Estilo	9
7. Estructura del código.....	11
7.1. Estructura tipo de un módulo PL/SQL	11
7.1.1. Bloques anónimos	11
7.2. Anidamiento de bloques	11
7.3. Funciones auxiliares	12
7.4. Uso de variables.....	12
8. Nomenclatura	13
8.1. Identificadores.....	13
8.1.1. Consideraciones Generales.....	13
8.1.2. Nomenclatura.....	13
8.2. Objetos.....	13
8.3. Acrónimos, abreviaturas y siglas permitidos	13
8.4. Alias	14
8.4.1. Normas para alias sobre tablas.....	14
8.4.2. Normas para alias sobre columnas.....	14

	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	



1. Introducción

En cualquier proyecto de desarrollo es importante emplear unas normas de codificación claras y homogéneas para todo el equipo de desarrollo. Particularmente importante es en el caso de proyectos de envergadura compuestos de un grupo numeroso de personas.

Estas normas no mejoran los programas en cuanto a rendimiento, eficacia o cumplimiento del objetivo del código, pero si facilita la comprensión y mantenimiento del código escrito en lenguaje PL/SQL, mejorando el tiempo empleado en las tareas de mantenimiento y corrección.

2. Objetivos y alcance

Las normas recogidas en el presente documento pretende crear un entorno de desarrollo homogéneo. El contenido de este documento es de aplicación en la construcción de cualquier componente que contenga código SQL y PL/SQL

 JUNTA DE ANDALUCÍA CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

3. Normas Generales.



Algunas de las directrices marcadas para el la codificación SQL y PL/SQL **pueden estar automatizadas** a través de la herramienta utilizada para el desarrollo, o por otras utilidades complementarias, por los que pueden “pasar a un segundo plano” durante la construcción del componente.

En estos casos, **antes de que el componente se considere finalizado** y se pueda a proceder a su implantación en cualquier entorno, éste **debe ser formateado correctamente** (apoyándose en las posibles utilidades de la herramienta de desarrollo).

3.1. Uso de mayúsculas y minúsculas

Todas el código utilizado ha de estar escrito en letras mayúsculas, salvo cadenas de texto y comentarios.



3.2. Líneas

- **Longitud de línea**

La longitud máxima se establece en 120 caracteres por línea.

- **Líneas en blanco**

No se permiten dos o más líneas en blanco consecutivas.

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas 06.02. Normas de Codificación SQL y PL/SQL	

4. Formato del Código.

4.1. Alineación e Identación.

Se establece una tabulación de tres caracteres. Estos han de ser espacios en blanco y no el carácter de tabulador.

4.1.1. Bloque anónimo.

Las palabras claves **Begin**, **Exception** y **End** se encontrarán alineadas, teniendo una indentación en las líneas internas. Dentro del bloque de excepciones las sentencia **When** aparece indentada respecto a **Exception**, y las sentencias que incluyen se indentan a su vez respecto a la palabra **When**.



```
BEGIN
  NULL;
EXCEPTION
  WHEN OTHERS THEN
    NULL;
END;
```

4.1.2. Condiciones.



Las palabras claves **If**, **Else**, **Elsif** y **End If** se encuentran alineadas en la misma columna para la misma sentencia, teniendo una indentación en las líneas internas. La palabra clave **Then** se pondrá a continuación de la condición.



```
IF <condiciones> THEN
  <sentencias>
ELSIF <condiciones> THEN
  <sentencias>
ELSE
  <sentencias>
END IF;
```

4.1.3. Bucles.

Las palabras claves **Loop** y **End Loop** se encuentran alineadas en la misma columna para la misma sentencia, teniendo una indentación en las líneas internas. Las palabras claves **While/For** y **End Loop** se encontrarán en la misma columna para la misma sentencia, teniendo una indentación en las líneas internas, la palabra clave **Loop** se pondrá a continuación de la condición.

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas 06.02. Normas de Codificación SQL y PL/SQL	

4.1.4. Sentencia SELECT.

Las distintas cláusulas (**Into, From, Where, Order By, Start With, Connect, Having, Group By...**) irán siempre al comienzo de una nueva línea, así como los operadores lógicos (**And, Or ...**).

Cada una de las cláusulas aparece justificada por la derecha. Los campos, variables, tablas y condiciones han de estar justificadas por la izquierda alineadas sobre la misma columna.



```

SELECT <campos>,
       <campos>
INTO <variables>,
    <variables>
FROM <tablas>,
    <tablas>
WHERE <condiciones>
      OPE <condiciones>
START WITH <expresión>
CONNECT BY <expresión>
ORDER BY <campos>

```

4.1.5. Sentencias Insert/Delete/Update.

Se deberá comenzar en una nueva línea en la cláusula (**Where, Set, Values**), de forma análoga a la sentencia **Select**.

4.1.6. Expresiones largas



Si una expresión es excesivamente larga se ha de cortar en líneas, de forma que el comienzo de todas las líneas estén en la misma columna. El corte ha de ser de forma que no se pierda la legibilidad.



```

SELECT 1
FROM IHCABACTADM CAA,
     IHGRACONPER IHG,
     IHRELADM RAD,
     IHPERSONAS PER
WHERE CAA.X_CABACTADM=:GLOBAL.IH_VG_X_CABACTADM#
      AND CAA.PER_X_PERSONA = PER.X_PERSONA
      AND CAA.APC_ACA_C_ACTADM = IH_FU_VALOR ('ARGP')
      AND CAA.X_CABACTADM = IHG.CAA_X_CABACTADM
      AND RAD.CAA_X_CABACTADM = PER.CAA_X_CABACTADM
      AND RAD.MIC_C_COLECTIVO= IH_FU_VALOR('CLAG')
      AND ( C_TIPSERGCP = IH_FU_VALOR('????')
      OR
          ( C_TIPSERGCP = IH_FU_VALOR('????')
            AND IH_FU_RADPTRES( RAD.CAA_X_CABACTADM )='N'
            AND NOT EXISTS (SELECT 1
                           FROM IHCABACTADM CAA1, IHTIPSITPER TSP
                           WHERE RAD.CAA_X_CABACTADM = CAA1.CAA_X_CABACTADM
                           AND TSP.CAA_X_CABACTADM = CAA1.X_CABACTADM
                           AND TSP.L_ULTSITPER = 'S'
                           AND TSP.TST_C_SITUACION = IH_FU_VALOR('????')
                           AND TST_C_TIPSIT <> IH_FU_VALOR('????')
                           )
            )
      )

```



 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	



```

SELECT 1 FROM IHCABACTADM CAA, IHGRACONPER IHG, IHRELADM RAD, IHPERSONAS PER
WHERE CAA.X_CABACTADM=:GLOBAL.IH_VG_X_CABACTADM# AND
IHG.CAA_X_CABACTADM=CAA.X_CABACTADM AND CAA.APC_ACA_C_ACTADM = IH_FU_VALOR ('ARGP')
AND CAA.PER_X_PERSONA = PER.X_PERSONA AND
RAD.CAA_X_CABACTADM=PER.CAA_X_CABACTADMRAD AND RAD.MIC_C_COLECTIVO=
IH_FU_VALOR('CLAG')
AND ( C_TIPSERGCP = 'A' OR ( C_TIPSERGCP = 'J'
AND IH_FU_RADPTRES( RAD.CAA_X_CABACTADM )='N' --NO TENGA PUESTO RESERVADO
-- PERSONA NO ESTE ACTIVO
AND NOT EXISTS (SELECT 1FROM IHCABACTADM CAA1, IHTIPSITPER TSP
WHERE RAD.CAA_X_CABACTADM = CAA1.CAA_X_CABACTADMRAD
AND TSP.CAA_X_CABACTADM = CAA1.X_CABACTADM
AND TSP.L_ULTSITPER = 'S' AND TSP.TST_C_SITUACION = '01' AND TST_C_TIPSIT <> '0003'))))

```

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

5. Comentarios.

5.1. Consideraciones Generales

La utilización correcta de comentarios favorece la comprensión del código y facilita por tanto las labores de corrección y reutilización de los componentes PL/SQL. Quedan fuera de este apartado los comentarios cabecera de objetos, caso particular que se trata específicamente en el siguiente apartado.

Consideraciones a tener en cuenta en la elaboración de comentarios:

- Los comentarios deben ser descriptivos de lo que se hace y no meras traducciones del código.
- No se permiten dos líneas de comentario en blanco consecutivas.
- No se comentan líneas de código. Si el código es sustituido por otro, las líneas de código han de ser borradas. Para ello se requerirá una copia del código anterior en un fichero a parte.

5.2. Dónde utilizarlos

Siempre que se haga uso de una constante se debe indicar la descripción de la misma.

5.3. Estilo de los comentarios





```
-- Este cursor obtiene los actos administrativos de la persona (P_X_PERSONA)
-- ***** Orden8888. WQRT. 11/11/03.
-- Se cambia el cursor para que recupere las cabeceras por f_creacion
-- descendente ya que antes no había orden ninguno. Además se incluye
-- el campo apc_aca_c_actadm
CURSOR CU_ACTOS IS
  SELECT X_CABACTADM, APC_ACA_C_ACTADM
  FROM IHCABACTADM
  WHERE PER_X_PERSONA = P_X_PERSONA
  ORDER BY F_CREACION DESC;
```



```
CURSOR CU_ACTOS IS
-- SELECT X_CABACTADM -- WQRT. 11/11/03
SELECT X_CABACTADM, APC_ACA_C_ACTADM
  FROM IHCABACTADM
 WHERE PER_X_PERSONA = P_X_PERSONA
  ORDER BY F_CREACION DESC; -- WQRT. 11/11/03
```

← Falta la descripción del cursor. Además se mantiene una línea de código comentada

← ¿Corresponde a una línea nueva o modificada?



 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	



<pre> CURSOR CU_ACTOS IS -- ***** -- WQRT. 11/11/03 -- SELECT X_CABACTADM SELECT X_CABACTADM, APC_ACA_C_ACTADM -- FIN 11/11/03 -- ***** FROM IHCABACTADM WHERE PER_X_PERSONA = P_X_PERSONA -- ***** -- WQRT. PONEMOS EL ORDER BY F_CREACION DESC. ORDER BY F_CREACION DESC; -- FIN WQRT. -- ***** </pre>	<p>← Cursor mal descrito y una línea de código comentada</p> <p>← Como en el caso anterior el comentario no aporta información</p> <p>← ¿Fin de qué?.</p>
--	---




<pre> FUNCTION FU_PERSONA (P_X_PERSONA IN IHPERSONAS.X_PERSONA%TYPE) RETURN VARCHAR2 IS V_NOMBRE VARCHAR2(120); CURSOR C_PERSONA IS SELECT NOMBRE ' ' APELLIDO1 ' ' APELLIDO2 FROM IHPERSONAS WHERE X_PERSONA = P_X_PERSONA; BEGIN /* SE ABRE EL CURSOR C_PERSONA */ OPEN C_PERSONA; /* SE RECUPERA EL NOMBRE DE LA PERSONA */ FETCH C_PERSONA INTO V_NOMBRE; /* SE CIERRA EL CURSOR */ CLOSE C_PERSONA; /* SE DEVUELVE EL VALOR RECUPERADO */ RETURN V_NOMBRE; END; /* FU_PERSONA */ </pre>	<p>← Ausencia de la descripción del cursor</p> <p>← Evidente</p> <p>← Evidente</p> <p>← Evidente</p> <p>← Evidente</p>
---	--

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

6. Documentación interna

6.1. Estilo



Las funciones, procedimientos y paquetes tendrán un bloque de comentario (documental) que seguirá el siguiente formato:

PLANTILLA DE DOCUMENTACIÓN INTERNA	
 <pre> /***** /* NOMBRE: el nombre de la función/procedimiento */ /* DESCRIPCION: lo que hace y no como lo hace */ /* PARAMETROS: */ /* P_1 : DESCRIPCION/VALORES PERMITIDOS E/S */ /* RETORNO: descripción del valor de retorno si es una */ /* función */ /* ----- */ /* ORDEN: 9999 FECHA: DD/MM/YYYY AUTOR: XXXX (SADIEL S.A.) */ /* DESCRIPCIÓN: ORIGINAL */ /* ----- */ /* ORDEN: 8888 FECHA: DD/MM/YYYY AUTOR: YYYY (SADIEL S.A.) */ /* DESCRIPCIÓN: CAMBIO POR ESTE U OTRO MOTIVO */ /* ----- */ *****/ </pre>	<div>Área de Creación</div> <div>Área de Modificaciones</div>

Hay que diferenciar dentro de la cabecera, el área de creación del objeto del área de modificaciones que le hayan afectado. En el caso de que las modificaciones afecten al número de parámetros o a su naturaleza se actualiza directamente en el área de creación.

En el área de modificaciones ha de reflejarse lo siguiente:

- Descripción de la nueva funcionalidad
- Descripción del motivo que provoca el cambio
- Número de orden
- Fecha en la que se realiza la modificación
- Iniciales de la persona que realiza la modificación

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

EJEMPLO: Función simple que no requiere mayor detalle de comentario

```

/*****
/* NOMBRE: FU_PERSONA */
/* DESCRIPCION: Recuperar el nombre de una persona*/
/* a partir de su P_X_PERSONA. */
/* PARAMETROS: */
/* P_X_PERSONA: Código de la persona */
/* RETORNO: Devuelve nombre y apellidos de */
/* la persona en una sola cadena con el */
/* formato NOMBRE+APELLIDO1+APELLIDO2 separado */
/* por un espacio. */
/* ORDEN FECHA AUTOR DESCRIPCIÓN */
/* ----- */
/* XXXX dd/mm/aaaa XXXX ORIGINAL */
*****/



FUNCTION FU_PERSONA
(P_X_PERSONA IN IHPERSONAS.X_PERSONA%TYPE)
RETURN VARCHAR2
IS
CURSOR CU_PERSONA IS
SELECT NOMBRE||' '||APELLIDO1||' '||APELLIDO2
FROM IHPERSONAS
WHERE X_PERSONA = P_X_PERSONA;

VL_NOMBRE VARCHAR2(120);

BEGIN
OPEN CU_PERSONA;
FETCH CU_PERSONA
INTO VL_NOMBRE;
CLOSE CU_PERSONA;
RETURN VL_NOMBRE;
EXCEPTION
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20000,'APL-...');
END; /* FU_PERSONA */


```



 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	


7. Estructura del código.

7.1. Estructura tipo de un módulo PL/SQL

	<pre> /***** /* NOMBRE: ... /* DESCRIPCION: ... /* . /* . /* . /* ----- /* ORDEN: ... /* . /* . /* . /***** ... BEGIN ... EXCEPTION ... END;</pre>	Cabecera. Área de Creación
		Cabecera. Área de Modificación
		Área de Declaraciones
		Cuerpo
		Área de Excepciones



7.1.1. Bloques anónimos

El orden de los elementos en un bloque de código anónimo es el siguiente:

	EJEMPLO/PLANTILLA: Estructura tipo de un bloque anónimo <pre> DECLARE <<DECLARACION DE CONSTANTES>> <<DECLARACION DE CURSORES>> <<DECLARACION DE VARIABLES LOCALES>> BEGIN <<SENTENCIAS>> EXCEPTION WHEN OTHERS THEN <<CONTROL DE LA EXCEPCION>> END;</pre>
---	---

7.2. Anidamiento de bloques

En el caso de bloques anónimos se permite el anidamiento de bloques, es decir un bloque anónimo puede contener otros bloques. En el caso de Funciones o Procedimientos el nivel de anidamiento se limita a uno, y generalmente para el control de errores.

	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	



7.3. Funciones auxiliares

El número de sentencias que forman una función auxiliar no puede ser superior al número de sentencias del bloque principal. En caso de que esto ocurra se recomienda su conversión en un paquete.

7.4. Uso de variables

No se permite el uso de variables y/o cursores definidos en el bloque principal dentro de las funciones auxiliares. Se deben utilizar parámetros en su lugar.

No se permite la redefinición de variables y/o cursores. En caso de anidamiento de bloques no pueden coincidir dos identificadores con el mismo nombre.

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
06. Instrucciones Técnicas		
06.02. Normas de Codificación SQL y PL/SQL		

8. Nomenclatura

8.1. Identificadores

8.1.1. Consideraciones Generales

- Deben tener un nombre claro y conciso.
- Se permite el uso del subrayado (_) en los identificadores, excepto al comienzo y al final del mismo.
- Se prohíbe la presencia de dos caracteres de subrayado juntos.

8.1.2. Nomenclatura

Se utilizará un prefijo indicativo del tipo de identificador. En el caso de cursores la longitud máxima se limita a quince caracteres.

Tipo de Objeto	Prefijo	Longitud máxima
Cursores	CU_	15
Parámetros	P_	-
Variables	V_	-
Variables Globales	VG_	-
Variables tipo registro (rowtype)	R_	-
Definiciones de tipo	T_	-

8.2. Objetos



Todos los objetos de base de datos estarán precedidos de un primer prefijo identificativo de la aplicación, IH, y un segundo en referencia a su tipo. Estarán limitados en su longitud máxima a quince caracteres.

Tipo de Objeto	Prefijo	Longitud máxima
Procedimiento	IH_PR_	15
Funciones	IH_FU_	15
Paquete	IH_PQ_	15

En el caso de Procedimientos y Funciones definidas en paquetes se puede omitir el prefijo de la aplicación.

8.3. Acrónimos, abreviaturas y siglas permitidos

Los siguiente tabla refleja algunas de las abreviaturas, siglas y acrónimos permitidos. No se reflejan en la tabla, pero se aceptan, las abreviaturas convencionales: DNI, NIF, ...

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas	
	06.02. Normas de Codificación SQL y PL/SQL	

Abreviatura	Descripción
Mto.	Mantenimiento
Cód.	Código
Abrev.	Abreviado/a
Apell.	Apellido
Nº	Número
NRP	Número Registro Personal
Gr.	Grupo
Gr. Prefte.	Grupo Preferente
Gr. Prof.	Grupo Profesional
Niv. Máx.	Nivel Máximo
Niv. Mín.	Nivel Mínimo
Cat.	Categoría
A.F.	Área Funcional
Plz.	Plaza
C. Específico	Tipo de Complemento Específico

8.4. Alias

Como práctica habitual en la implementación de sentencias *SQL* no se debe abusar del uso de alias. Los casos en los que son aceptados, y en ciertos casos recomendados, e incluso obligatorios se corresponden con:

- Duplicidad de tablas.
- Duplicidad de nombres de columnas.
- Columnas sin nombre provenientes de operaciones.
- Subconsultas.
- Columnas procedentes de consultas con operadores de conjunto.
- Consultas con un código muy largo que se puede simplificar con alias.

Los alias siempre se deberán usar de forma normalizada, evitando en todo caso el uso de un alias de elección libre por parte del desarrollador.

8.4.1. Normas para alias sobre tablas



En caso de utilizarse alias de tabla, dicho alias se corresponderá con el definido en el modelo de datos en Designer.

Si se repite un mismo alias/tabla se numerará o se pondrá un sufijo identificativo:

Ejemplos: CAA1, CAA2 ; EO_PADRE, EO_HIJA

8.4.2. Normas para alias sobre columnas

Como práctica habitual en la implementación de sentencias *SQL*, los alias de columnas son elementos que no deben ser utilizados, si bien la utilización de columnas con nombre repetido, la inclusión de fórmulas o por definir el nombre de columnas procedentes de

 JUNTA DE ANDALUCÍA <small>CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</small>	SISTEMA DE ASEGURAMIENTO DE LA CALIDAD	 SIRhUS
	06. Instrucciones Técnicas 06.02. Normas de Codificación SQL y PL/SQL	

consultas con operadores de conjunto (UNION, etc) pueden hacer recomendable su utilización en aras de un código más legible.

A la hora de elegir un alias para una columna se tendrán en cuenta las siguientes normas:

8.4.2.1. Columnas procedentes de FOREIGN KEYS

Sólo en los casos en los que se produzca duplicidad se permite el uso de alias.

- Si la duplicidad se produce entre columnas de tablas diferentes se usará un alias formando por el nombre de la tabla, un guión bajo y el nombre de la columna: <Tabla>_<Columna>.
- Si la duplicidad se produce entre columnas de la misma tabla se usará un alias formando por el nombre de la columna, seguido de un numeral, separado por un guión bajo.: <Columna>_<N>.

8.4.2.2. Columnas procedentes de fórmulas

- En fórmulas simples (TO_CHAR(),SUM(),AVG,...) se usará el alias formado por el nombre de la función, seguido de la columna a la que afecta, separado por un guión bajo.



Fórmulas simples
SELECT SUM(importe) sum_importe, ...



- En fórmulas complejas, con varias operaciones o que no den lugar a un alias significativo, se usará un alias descriptivo de la fórmula implementada.



Fórmulas complejas
SELECT SUBSTR(casilla_1,1,2)/100 retenido, ...

8.4.2.3. Columnas de consultas con operaciones de conjunto.

Cuando se utilizan operadores de conjunto en consultas, es aconsejable incluir un alias que aclare el significado de las columnas que sean resultado de tablas diferentes. Este alias tendrá que aportar el máximo significado posible.

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE JUSTICIA Y ADMINISTRACIÓN PÚBLICA</p>	<p>SISTEMA DE ASEGURAMIENTO DE LA CALIDAD</p>	 <p>SIRhUS</p>
	<p>06. Instrucciones Técnicas</p>	
	<p>06.02. Normas de Codificación SQL y PL/SQL</p>	



Solución para consultas con operadores de conjuntos
<pre> CURSOR c_expedientes (pc_desde DATE, pc_hasta DATE) IS SELECT nombre nombre, apellido1 ' ' apellido2 apellidos FROM ihpersonas WHERE f_nacimiento BETWEEN p_desde AND p_hasta UNION SELECT nombre nombre, apellido1 ' ' apellido2 apellidos FROM ihpersonas WHERE f_nacimiento BETWEEN p_desde AND p_hasta UNION SELECT nombre nombre, apellido1 ' ' apellido2 apellidos FROM ihpersonas WHERE f_nacimiento BETWEEN p_desde AND p_hasta ORDER BY nombre, apellidos; </pre>