

Excepcions. Activitat 1

Aquesta activitat mostra diferents exemples que fan servir excepcions.

Tots els exemples compilen correctament però tenen errors de lògica.

Alguns exemples utilitzen taules de l'esquema **HR** (o **HR2**).

Cada exemple:

- Mostra **un programa que no és del tot correcte**.
- Pregunta com es comporta el programa amb alguns valors d'entrada concrets.
- Demana modificar el programa per corregir errors o implementar millores.

Per treballar l'activitat:

- Primer, tracta de respondre les preguntes.
- Després, executa el programa i comprova que fa en cada cas.
- Per últim, implementa les modificacions demanades.
- Com més aprendràs és fent proves i modificant els programes.

Pots fer servir la documentació següent:

- PDF sobre excepcions al Moodle
- Excepcions del sistema: [PL/SQL Exceptions](#)
- Excepcions d'usuari. Llançar excepcions: [PL/SQL Raise Exceptions](#)
- Documentació d'Oracle: [Handling PL/SQL Errors](#)

També et serà útil recordar l'ús d'excepcions que hem fet fins ara:

- Comprovar si una divisió dona un resultat o provoca una excepció.
- Comprovar si el valor d'una variable de text es pot convertir o no en un valor numèric.
- Buscar el nom d'un empleat a partir del seu cognom. Si no n'hi ha empleats amb aquest cognom o si n'hi ha més d'un, mostrar un missatge d'error.
- Crear una funció per comprovar si el text que rep com a paràmetre és o no un import positiu.
- Crear una funció per comprovar si el text que rep com a paràmetre té o no el format dels NIF dels espanyols residents majors d'edat.

E1. Clàusula **when others**. Funcions **sqlcode** i **sqlerrm**.

```
/*
  Mostrar el rang salarial d'un job_id determinat
*/
accept vh_job_id prompt 'job_id a buscar?'
declare
  v_job      jobs%rowtype;
  v_job_id   jobs.job_id%type := '&vh_job_id';
begin
  --v_job_id := '&vh_job_id';
  select *
  into v_job
  from jobs
  where job_id = v_job_id;
  dbms_output.put_line('&vh_job_id: cobra de '
    || v_job.min_salary || ' a ' || v_job.max_salary || ' USD');
exception
  when others then
    dbms_output.put_line(sqlcode || ' | ' || sqlerrm);
end;
```

Què passa si executem el programa amb un **vh_job_id**...

- que no existeixi a la taula **jobs**?
- que tingui 11 caràcters? (job_id és un varchar2(10))

Opcions:

- El programa acaba amb error per una excepció **no_data_found** no capturada.
- El programa acaba amb error per una excepció **value_error** no capturada.
- El programa captura l'excepció i escriu «-4056 | ORA-04056: others».
- El programa captura l'excepció i escriu «-6502 | ORA-06502: PL/SQL: numeric or value error: character string buffer too small».
- El programa captura l'excepció i escriu «100 | ORA-01403: no data found».

Modifica el programa traient el **when others** i capturant per separat cada una de les dues excepcions que es poden produir.

```
/* mostrar el rang salarial d'un job_id determinat */

accept vh_job_id prompt 'job_id a buscar?'
declare
  v_job      jobs%rowtype;
  v_job_id   jobs.job_id%type := '&vh_job_id';
begin
  -- v_job_id := '&vh_job_id';
  select *
  into v_job
  from jobs
  where job_id = v_job_id;
  dbms_output.put_line('&vh_job_id: cobra de '
    || v_job.min_salary || ' a ' || v_job.max_salary ||
'USD');
```

```
exception
  when no_data_found then
    dbms_output.put_line('job no trobat: ' ||
v_job_id);
    dbms_output.put_line(sqlcode || ' - ' || sqlerrm);

  when value_error then
    dbms_output.put_line('Valor fora de rang: ' ||
v_min_salary);
    dbms_output.put_line(sqlcode || ' - ' || sqlerrm);

  when others then
    dbms_output.put_line(sqlcode || ' - ' || sqlerrm);

end;
```

E2. Excepcions més habituals.

```
/*
Mostrar el job_title amb un min_salary determinat. Si no n'hi ha cap, si n'hi
ha més d'un, o si l'usuari introdueix un valor no numéric, o un valor numéric
fora de rang, mostrar un missatge d'error adient.
*/
accept vh_min_sal prompt 'min_salary a buscar?'
declare
  v_min_sal varchar2(4);
  v_job_title varchar2(30);
begin
  v_min_sal := '&vh_min_sal';
  select job_title
  into v_job_title
  from jobs
  where min_salary = v_min_sal;
  dbms_output.put_line(v_job_title
  || ' cobra un sou mínim de ' || v_min_sal || ' USD');
exception
  when too_many_rows then
    dbms_output.put_line('Error: Més d'un job amb aquest min_salary.');
```

```
    dbms_output.put_line(sqlerrm);
  when value_error then
    dbms_output.put_line('Error: Valor fora de rang o no convertible en número.');
```

```
    dbms_output.put_line(sqlerrm);
end;
```





Segons aquestes dades de la taula **jobs**:

Worksheet

Query Builder

```
select min_salary, job_title, length(job_title) len_job_title
from jobs
where min_salary in (2000, 2500, 4500, 8000, 15000)
order by min_salary, job_title;
```

Query Result x

    SQL | All Rows Fetched: 5 in 0.007 seconds

	MIN_SALARY	JOB_TITLE	LEN_JOB_TITLE
1	2500	Purchasing Clerk	16
2	2500	Shipping Clerk	14
3	4500	Public Relations Representative	31
4	8000	Purchasing Manager	18
5	15000	Administration Vice President	29

Què passa si executem el programa amb **vh_min_sal** igual a...?

vh_min_sal	Excepció (sí o no)?	Quina (nom)?	On es produeix?	Es captura (sí o no)?
2000	sí	no_data_found	A la select	no
2500	sí	too_many_errors	where	si
4500				
8000				
15000				
5500800				
XX				

De vegades és difícil saber si un programa llançarà una excepció **value_error** o una excepció **invalid_value**, no et preocupis gaire per això.

Modifica el programa per declarar de forma correcta les variables fent servir **%type**, inicialitzar de forma correcta la variable **v_min_sal** i capturar per separat cada una de les excepcions que es poden produir. Executa el teu programa amb cada un dels valors proposats. Comprova que capturi les excepcions i mostri els missatges d'error adients.

E3. Excepcions d'usuari. Funcions d'agregació: **avg**, **count**, **sum**, etc.

Per declarar una excepció d'usuari fem servir el tipus **exception**:

```
nom_excepcio exception;
```

Per llançar una excepció d'usuari o de sistema fem servir la sentència **raise**:

```
raise nom_excepcio;
```

Documentació:

[OracleTutorial: PL/SQL Raise Exceptions](#)

Exemple:

```
/*
  Mostrar el sou mitjà dels empleats d'un department_id determinat
  Mostrar missatges d'error diferents si el department_id és negatiu
  o si no té empleats.
  Fer servir excepcions d'usuari per llançar els missatges d'error.
*/
accept vh_dep_id prompt 'department_id a buscar?'
declare
  v_sou_mitja      employees.salary%type;
  v_dep_id         departments.department_id%type;
  e_id_negatiu     exception;
  e_id_inexistent exception;
begin
  v_dep_id := &vh_dep_id;
  if v_dep_id < 0 then
    raise e_id_negatiu;
  end if;
  select avg(salary)
    into v_sou_mitja
  from employees
 where department_id = v_dep_id;
  dbms_output.put_line('Sou mitjà: ' || to_char(v_sou_mitja, 'fm99G999') || ' USD');
exception
  when e_id_negatiu then
    dbms_output.put_line('error: department_id negatiu');
  when no_data_found then
    dbms_output.put_line('error: departent_id sense empleats');
end;
```

Què passa si executem el programa amb un valor enter de **vh_dep_id**...

- que sigui negatiu?
- que existeixi a la taula **employees**?
- que no existeixi a la taula **employees**?

Modifica el programa per que faci allò que diu la seva especificació.

Has de comprovar si no hi ha cap empleat que pertanyi al departament **vh_dep_id** i, en aquest cas, llançar l'excepció **e_id_inexistent**, capturar-la a la secció **exception** i mostrar un missatge d'error adient.

L'excepció del sistema **no_data_found** no s'ha de produir mai.

E4. Assignar codis d'error. Llançar excepcions amb missatges personalitzats.

Per assignar un codi d'error a una excepció d'usuari fem servir, a la secció **declare**, la directiva de compilació:

```
pragma exception_init(nom_excepcio, codi_error);
```

Per llançar una excepció d'usuari amb un missatge d'error personalitzat fem servir el procediment:

```
raise_application_error(codi_error, missatge_error);
```

Els codis d'error reservats per a excepcions personalitzades van del -20.000 al -20.999.

Documentació:

[OracleTutorial: RAISE_APPLICATION_ERROR](#)

Exemple:

```
/*
Validar una matrícula espanyola amb el sistema de numeració europeu que va
entrar en vigor l'any 2000.
En aquest sistema les matrícules tenen 4 dígits seguits de 3 lletres excloses
les vocals i les consonants "Ç", "Ñ" i "Q".
Mostrar un missatge indicant si la matrícula és vàlida o quin error té.
*/
accept vh_matricula prompt 'matrícula a validar?'
declare
c_digits constant varchar2(10) := '0123456789';
c_lletres constant varchar2(20) := 'BCDFGHJKLMNPSTWVXYZ';
v_matricula varchar2(7);
e_valor_nul exception;          pragma exception_init(e_valor_nul, -20001);
e_digit_invalid exception;      pragma exception_init(e_digit_invalid, -20002);
-- e_lletra_minuscula exception;
e_lletra_invalida exception;
begin
v_matricula := '&vh_matricula';
if v_matricula is null then
raise e_valor_nul;
end if;
for i in 1..4 loop
if instr(c_digits, substr(v_matricula, i, 1)) = 0 then
raise_application_error(-20002, 'Dígits invàlids en la matrícula');
end if;
end loop;
if v_matricula != upper(v_matricula) then
raise_application_error(-20003, 'Lletres minúscules en la matrícula');
end if;
for i in 5..7 loop
if instr(c_lletres, substr(v_matricula, i, 1)) = 0 then
raise e_lletra_invalida;
end if;
end loop;
dbms_output.put_line('la matrícula és vàlida');
exception
when e_digit_invalid then
dbms_output.put_line('e_digit_invalid | ' || sqlcode || ' | ' || sqlerrm);
when others then
dbms_output.put_line(sqlcode || ' | ' || sqlerrm);
end;
```

Què escriu el programa si l'executem amb **vh_matricula** igual a...?

- 908BKB
- 0668HUC
- B2379YJ
- GI5701WR
- 9297RXB
- (una cadena buida)

Opcions:

- -20001 | ORA-20001:
- -20001 | e_valor_nul
- -20001 | User-Defined Exception
- -20002 | ORA-20002: Dígits invàlids en la matrícula
- -20003 | ORA-20003: Lletres minúscules en la matrícula
- -29999 | User-Defined Exception
- -29999 | e_lletra_invalida
- -6502 | ORA-06502: PL/SQL: numeric or value error: character string buffer too small
- -6502 | ORA-06502: PL/SQL: numeric or value error: character to number conversion error
- -6502 | ORA-06502: PL/SQL: numeric or value error: number precision too large
- 1 | User-Defined Exception
- 1 | e_lletra_invalida
- e_digit_invalid | -20002 | ORA-20002: Dígits invàlids en la matrícula
- e_lletra_minuscula | -20003 | ORA-20003: Lletres minúscules en la matrícula
- la matrícula és vàlida

Modifica el programa per fer que cada excepció d'usuari tingui un codi d'error propi i es llanci amb un missatge d'error personalitzat.

Per exemple, l'excepció **e_lletra_invalida** podria tenir el codi d'error -20004 i llançar-se amb el missatge «Lletres invàlides en la matrícula».

No cal que capturis cada excepció per separat, captura-les totes amb el **when others**.

E5. Propagació d'excepcions en blocs niuats.

Quan un bloc niuat («bloque anidado», «nested block») no captura una excepció, aquesta es propaga al bloc dintre del qual hi és.

Quan una funció o un procediment no captura una excepció, aquesta es propaga al bloc que ha cridat a la funció o al procediment.

Quan una excepció arriba al programa principal, si el programa principal no la captura, l'excepció ja no es pot propagar enlloc més, llavors provoca un error d'execució i el programa finalitza.

Documentació:

[OracleTutorial: PL/SQL Exception Propagation](#)

Exemple: Funció **sol_eq_quad**.

```
/*
  Calcular les solucions d'una equació quadràtica expressada en forma estàndard
  amb coeficients p_a, p_b i p_c.
  https://ca.wikipedia.org/wiki/Equaci%C3%B3_de_segona_grau
*/
create or replace procedure sol_eq_quad(
  p_a in number, p_b in number, p_c in number
  , p_r1 out number, p_r2 out number
  , p_msg out varchar2
)
is -- bloc 1
begin
  p_msg := 'begin';
  begin -- bloc 1.1
    p_r1 := (-p_b + sqrt(p_b * p_b - 4 * p_a * p_c)) / (2 * p_a);
  exception
    when value_error then p_msg := p_msg || ' | bloc 1.1: VE';
    when zero_divide then p_msg := p_msg || ' | bloc 1.1: ZD';
  end;
  begin -- bloc 1.2
    p_r2 := (-p_b - sqrt(p_b * p_b - 4 * p_a * p_c)) / (2 * p_a);
  exception
    when value_error then p_msg := p_msg || ' | bloc 1.2: VE';
  end;
  p_msg := p_msg || ' | end';
exception
  when value_error then p_msg := p_msg || ' | bloc 1: VE';
  when zero_divide then p_msg := p_msg || ' | bloc 1: ZD';
end;
```

Quin és el valor de sortida de **p_msg** si cridem a la funció **sol_eq_quad** amb els paràmetres d'entrada..?

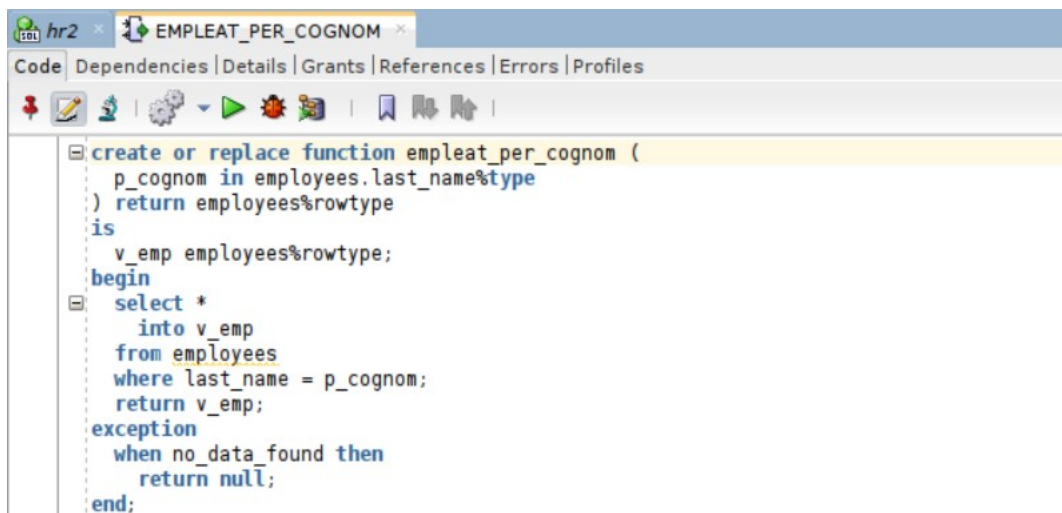
Paràmetres d'entrada			Paràmetres de sortida (valor retornat o error que es produeix)		
p_a	p_b	p_c	p_r1	p_r2	p_msg
0	6	1	divisió per zero	divisió per zero	
5	6	1	-0,2	-1	begin end
5	2	1	arrel d'un negatiu	arrel d'un negatiu	
2	-4	2	1	1	

E6. Propagació d'excepcions en funcions.

Programa principal:

```
/*
  Mostrar nom, cognom i salari de l'empleat amb un cognom determinat.
  Mostrar un missatge d'error si l'empleat/ada no existeix o si n'hi ha més d'un/a.
*/
accept vh_cognom prompt 'Cognom a buscar?'
declare
  v_cognom employees.last_name%type;
  v_emp employees%rowtype;
begin
  v_cognom := '&vh_cognom';
  v_emp := empleat_per_cognom(v_cognom);
  dbms_output.put_line('Nom: ' || v_emp.first_name);
  dbms_output.put_line('Cognom: ' || v_emp.last_name);
  dbms_output.put_line('Salari: ' || v_emp.salary);
exception
  when no_data_found or too_many_rows then
    dbms_output.put_line('Error: Cap o més d'un/a empleat/da amb aquest cognom.');
```

Funció **empleat_per_cognom**:



```
create or replace function empleat_per_cognom (
  p_cognom in employees.last_name%type
) return employees%rowtype
is
  v_emp employees%rowtype;
begin
  select *
  into v_emp
  from employees
  where last_name = p_cognom;
  return v_emp;
exception
  when no_data_found then
    return null;
end;
```

Què passa si executem el programa amb **vh_cognom** igual a...?

- Tazi (cap empleat/da té aquest cognom)
- Vargas (només un empleat té aquest cognom)
- Smith (més d'un/a empleat/da té aquest cognom)

Opcions:

- El programa acaba amb error perquè la funció no captura l'excepció `no_data_found`.
- El programa acaba amb error perquè la funció no captura l'excepció `too_many_rows`.
- El programa acaba amb error perquè `v_emp` no té valor.
- El programa escriu «Error: Cap o més d'un/a empleat/da amb aquest cognom.»
- El programa escriu «Nom: », «Cognom: » i «Salari: » amb els valors corresponents.
- El programa escriu «Nom: », «Cognom: » i «Salari: » però sense cap valor.

Modifica la funció i el programa principal per que la captura d'excepcions es faci només al programa principal.