Docente: **Matteo Re**

UNIVERSITÀ
DEGLI STUDI
DI MILANO

C.d.l. Beni culturali

A.A. 2019-2020 semestre I

Metodi e linguaggi per il trattamento dei dati

p4 Interrogazione diretta di banche dati biologiche - SQL

Obiettivi

- **Interrogazione diretta di banche dati biologiche**
  - Accesso mediante Perl
  - Linguaggio SQL
- **Database relazionali**
  - Struttura (db schema) Ensembl database
  - API Ensembl
  - Estrazione di annotazioni
  - Estrazione di sequenze

Linee guida

- **Il livello di complessità di questa esercitazione è medio-alto**
  - Cercate di risolvere il problema dopo aver compreso gli schemi dai database presentati
  - I template script di questa esercitazione sono estremamente semplici … non fatevi ingennare da questa apparente semplicità **la difficoltà dell'esercizio risiede nella necessità di costruire le interrogazioni in linguaggio SQL** e di itegrarle in maniera opportuna negli script. Come sempre il codice che mi invierete DEVE essere commentato (in questo caso il commento riguarderà principalmente le query SQL).

- **Modalità di svolgimento dell'esercitazione:**
  - Nessun file da scaricare questa volta … lo script di base per effettuare le query SQL è molto contenuto ed è riportato in queste slide.
  - Lo stesso vale per gli esercizi sulle API Ensembl core (trovate molti più esempi risolti mediante le API che mediante SQL… Questo dipende dal fatto che la difficoltà intrinseca degli esercizi SQL sta nella necessità **di dover esplorare lo schema della banca dati Ensembl**).

Tipi di banche dati biologiche:

Collettori primari:

Sequenze sottomesse direttamente dai laboratori di ricerca alle banche dati Genbank, DDBJ ed EMBL. Qualità bassa, a volte contengono errori di annotazione.

Banche dati secondarie:

Le informazioni contenute in queste banche dati sono curate manualmente: qualità superiore. Spesso sono banche dati specializzate nel senso che contengono un solo tipo di informazione (seq. proteiche, seq. di trascritti, …).

Banche dati associate a progetti genomici:

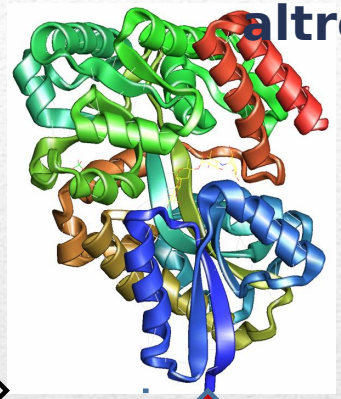
Le sequenze genomiche sono un tipo di dato molto particolare. Esse si prestano ad essere annotate a diversi livelli. A causa di questa caratteristica la loro annotazione richiede l'utilizzo di informazioni derivanti da un numero consistente di banche dati esterne. Come conseguenza le banche dati associate a progetti di annotazione genomica sono gli strumenti di elezione per **INTEGRARE** il contenuto di altre banche dati in modo da ottenerne una **rappresentazione unitaria**.

Tipi di dati biologici (solo alcuni)

- **Livello dei trascritti** misurati in particolari condizioni: esistono siti dedicati a collezioni di esperimenti microarray (es. NCBI Gene Expression Omnibus (NCBI GEO), http://www.ncbi.nlm.nih.gov/geo/)
- **Annotazione funzionale di proteine**: «funzionale» viene utilizzato come termine a «**basso**» livello, annotazione di una sequenza proteica **residuo per residuo**. Molti tipi di annotazione: siti di fosforilazione, presenza di ponti disolfuro, struttura secondaria della proteina, struttura 3D della proteina. Sito di riferimento è una banca dati che integra le informazioni di diverse banche dati: Uniprot (Universal Protein Resource, http://www.uniprot.org/) .
- **Annotazione funzionale di geni:** «funzionale» viene utilizzato come termine ad «**alto**» livello. Creazione di vocabolari controllati a partire da materiale reperibile in **LETTERATURA**. Team di curatori assegnano ogni gene ai termini dei vocabolari (**ontologie**). Sito di riferimento: Gene Ontology (http://www.geneontology.org/) .
- **Variabilità genetica:** Database dedicati a SNP (es. NCBI dbSNP) e a progetti su vasta scala (HapMap). Esistono inoltre databases dedicati a studi di associazione genome-wide (es GWAS central) http://www.gwascentral.org/index.

E MOLTI ALTRI …

Esistono molti tipi di banche dati … perché quelle associate a progetti di annotazione genomica dovrebbero essere considerate più «importanti» di altre?



fuctional annotation

post tanslational editing

post transcriptional editing

folding

translation

transcription

genome

regulatory region

regulatory motifs

Proteins DB

cross links

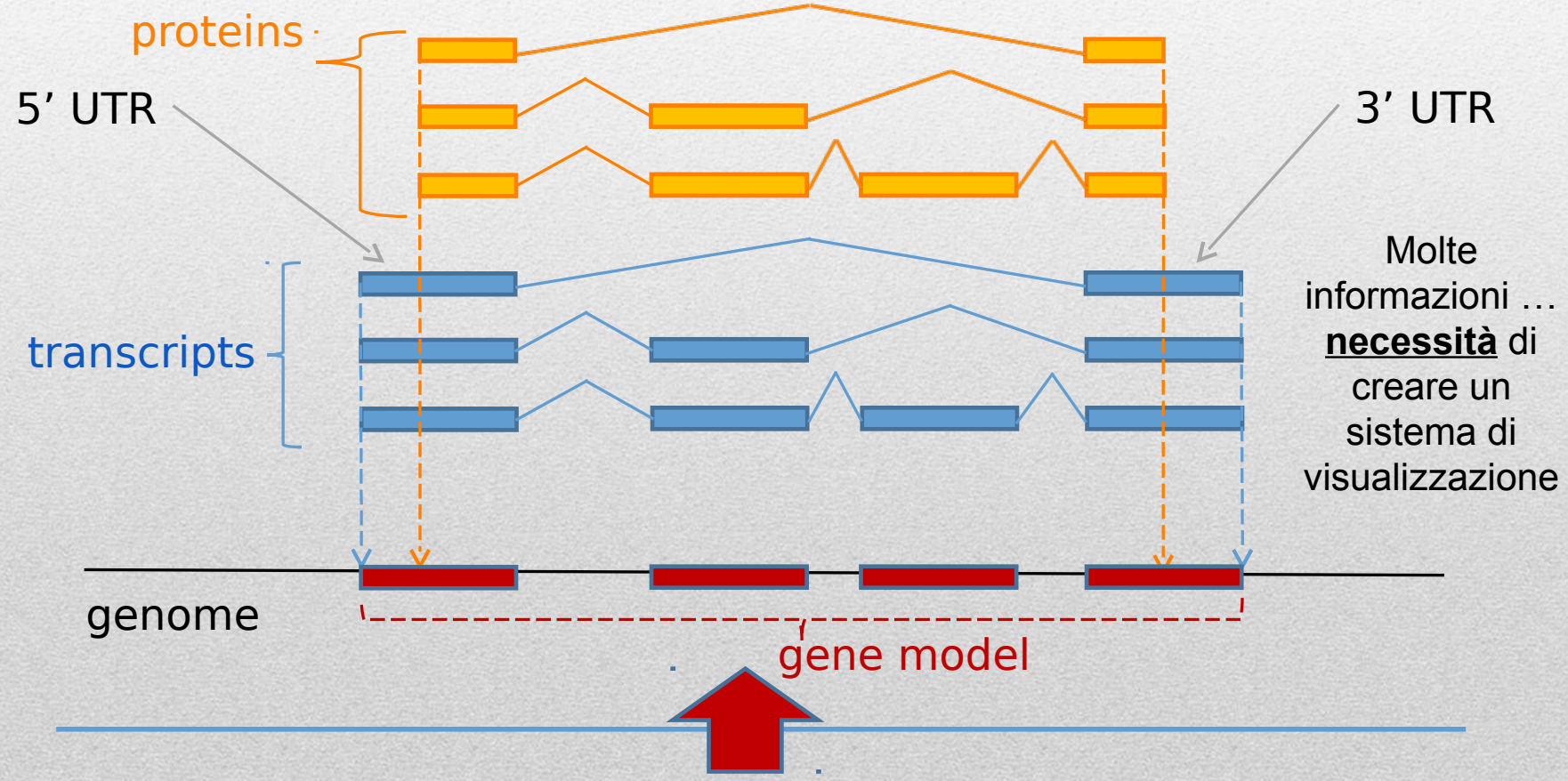
Transcripts DB

alignment

alignment

Annotazione in coordinate genomiche

Se «proiettiamo» tutte le informazioni disponibili (seq. espresse, seq. proteiche, motivi regolatori ecc.) sul genoma rendiamo tali informazioni più semplici da consultare perché il genoma assume il ruolo di elemento di riferimento!



Annotazione in coordinate genomiche

BROWSERS GENOMICI

- **Ne esistono diversi:** Principalmente 3, NCBI map viewer (http://www.ncbi.nlm.nih.gov/projects/mapview/), Ensembl (http://www.ensembl.org/index.html) e UCSC genome browser (http://genome.ucsc.edu/) .
- **Presentano le stesse informazioni, ma in modo diverso**: tutti e tre permettono di trovare la posizione genomica di una sequenza (mediante allineamento o ricerca per parola chiave) e di visualizzare la regione genomica associata.
- **I dati contenuti nei browser genomici dipendono dal contenuto di altre banche dati:** necessità di aggiornare i dati molto spesso. Ensembl viene aggiornato mensilmente .
- **Produzione dei dati di annotazione genomica:** E' un processo cosatoso dal punto di vista delle risorse di calcolo (allineamento di intere banche dati di sequenze al genoma). I principali browser genomici contengono più di un genoma (in realtà contengono molti genomi). E' un processo basato su **pipeline di annotazione automatizzate**.

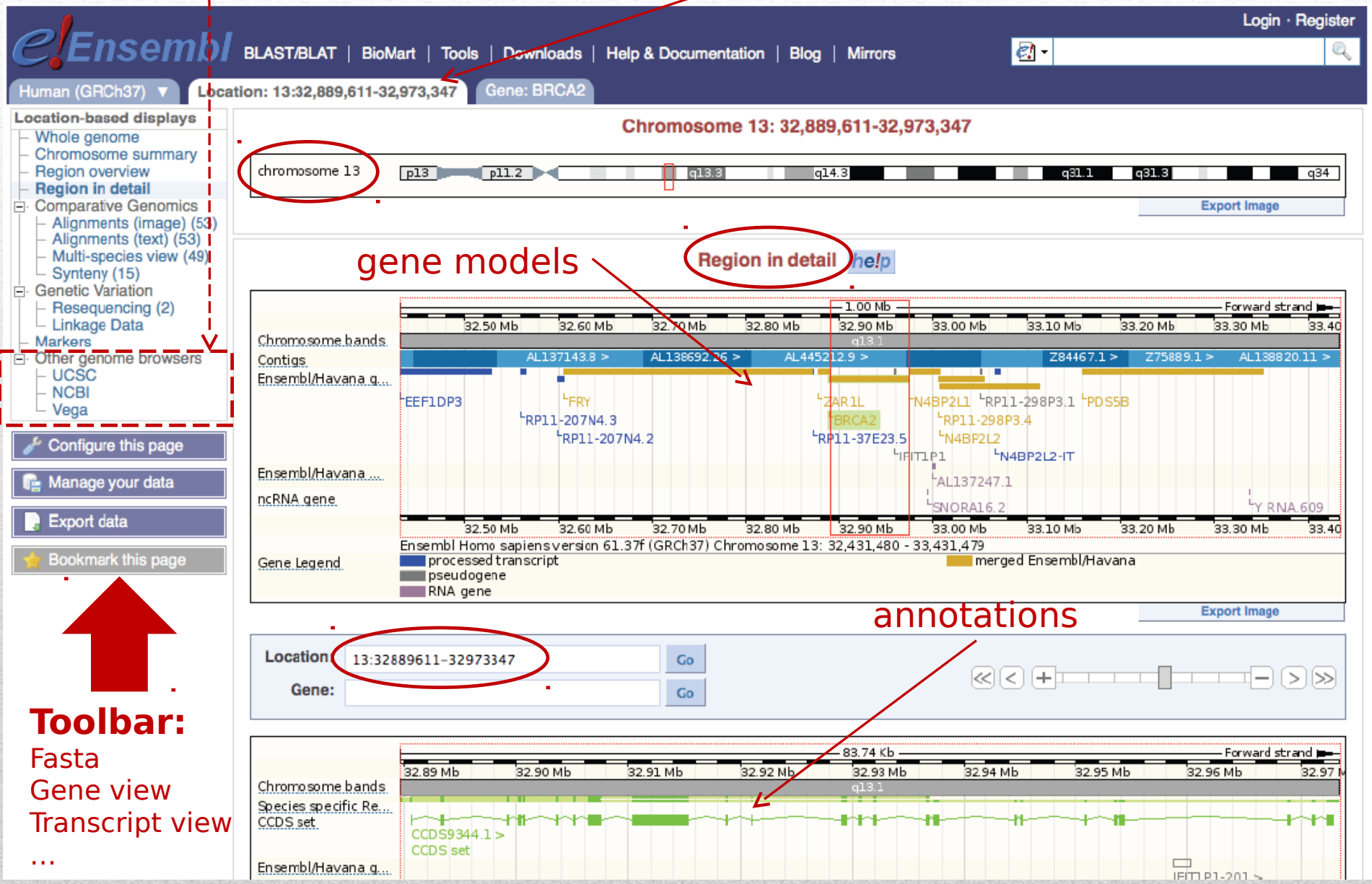
Ensembl

Caratteristica specifica di Ensembl :

contiene **modelli** di geni (altri browser utilizzano come entità fondamentale il trascritto o, comunque, la «sequenza allineata al genoma»)

external links

Ensembl: Genome view



Toolbar:

Fasta

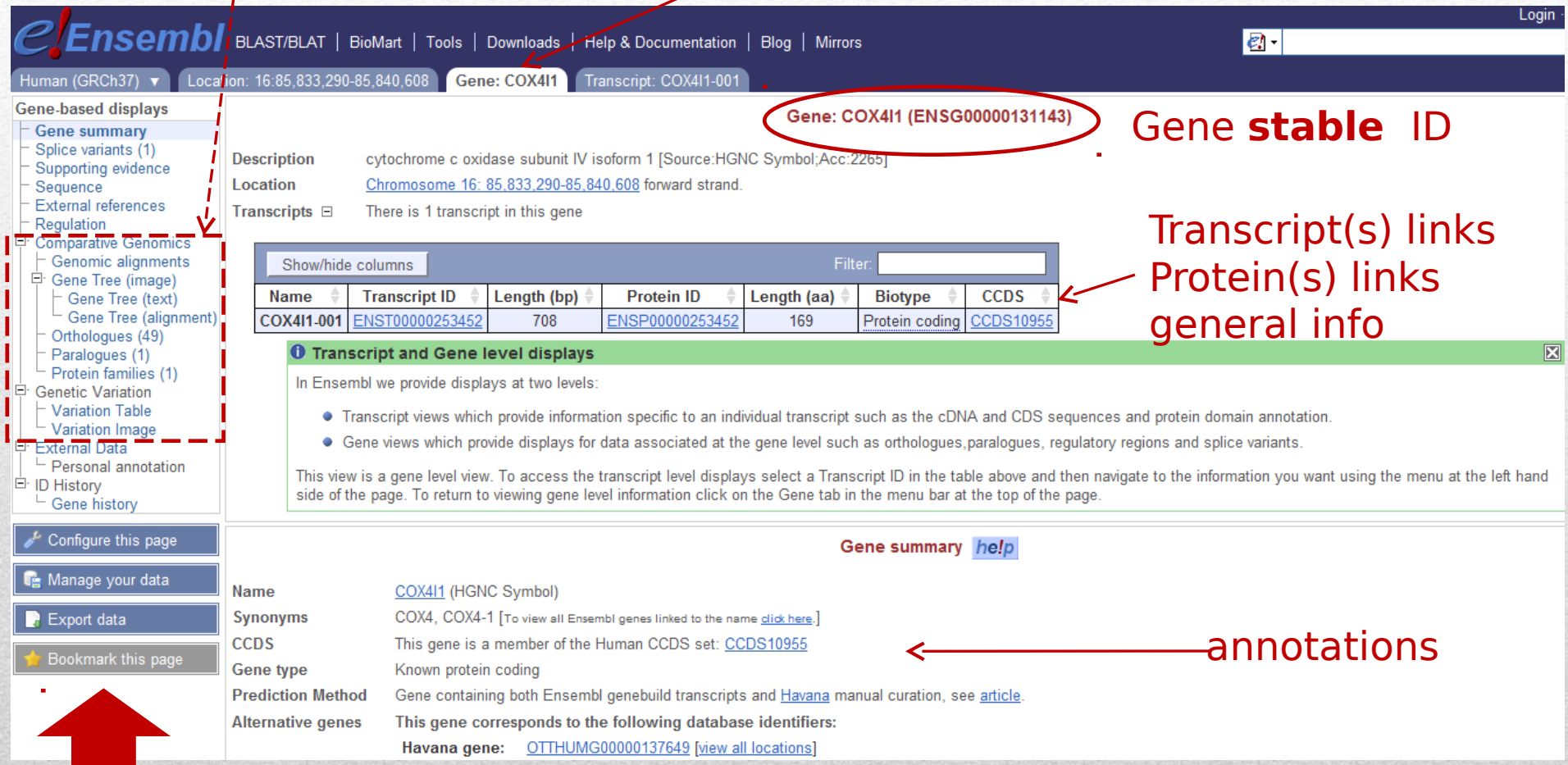
Gene view

Transcript view

...

external links

Ensembl: Gene view



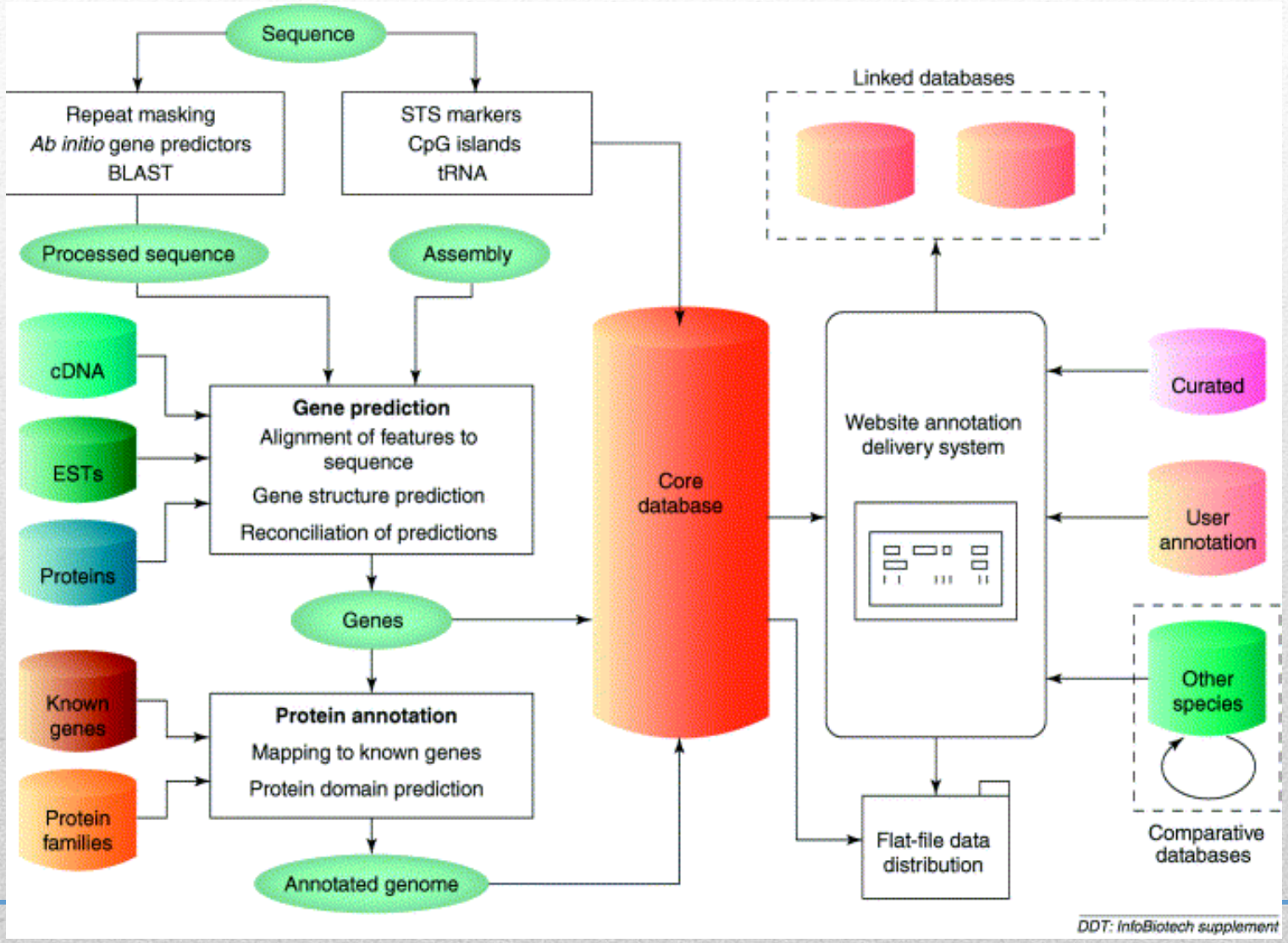
Gene **stable** ID

Transcript(s) links
Protein(s) links
general info

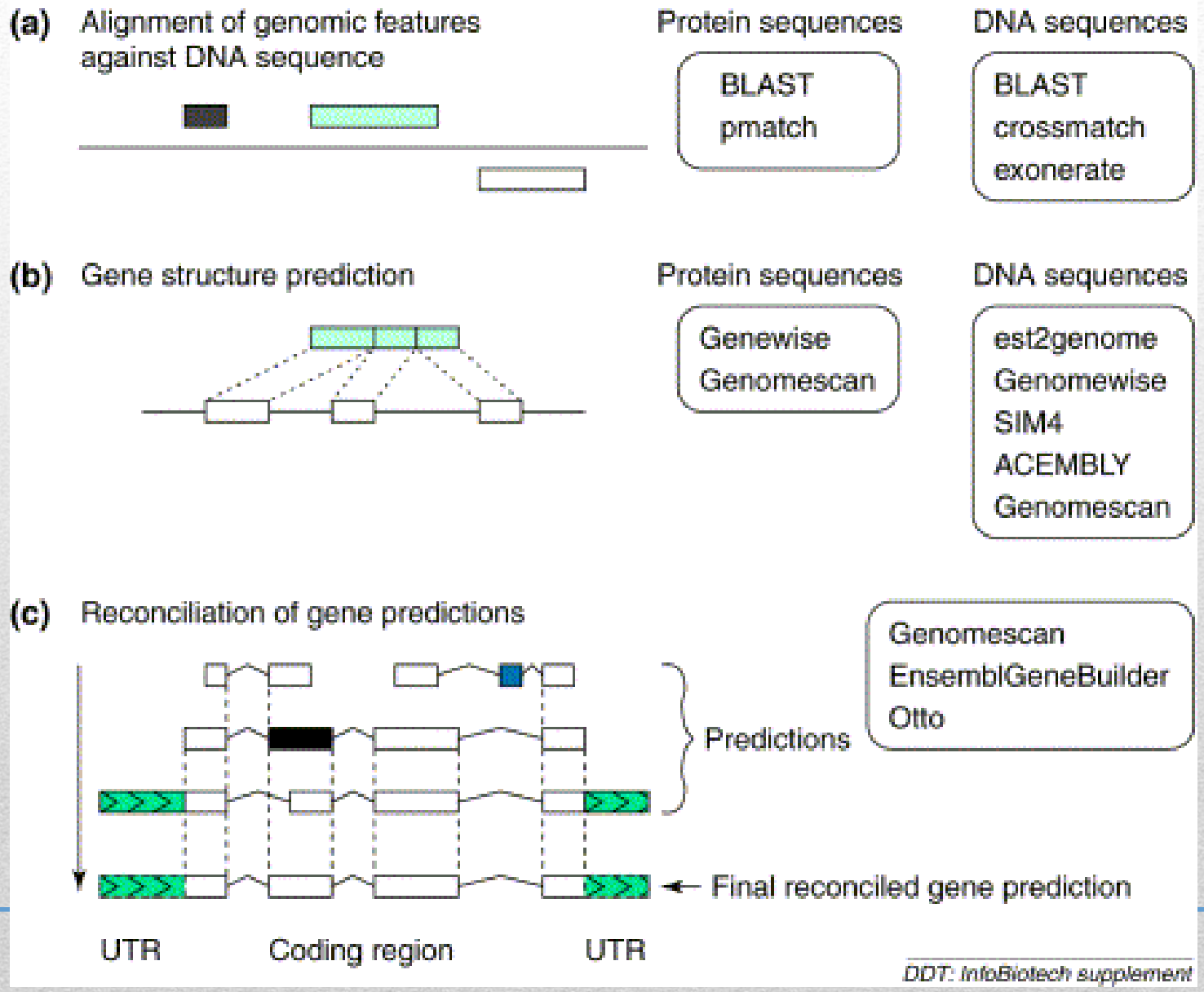
annotations

Toolbar:
Export seq.
Export annot.
...

Automatizzazione del processo di annotazione di una sequenza genomica

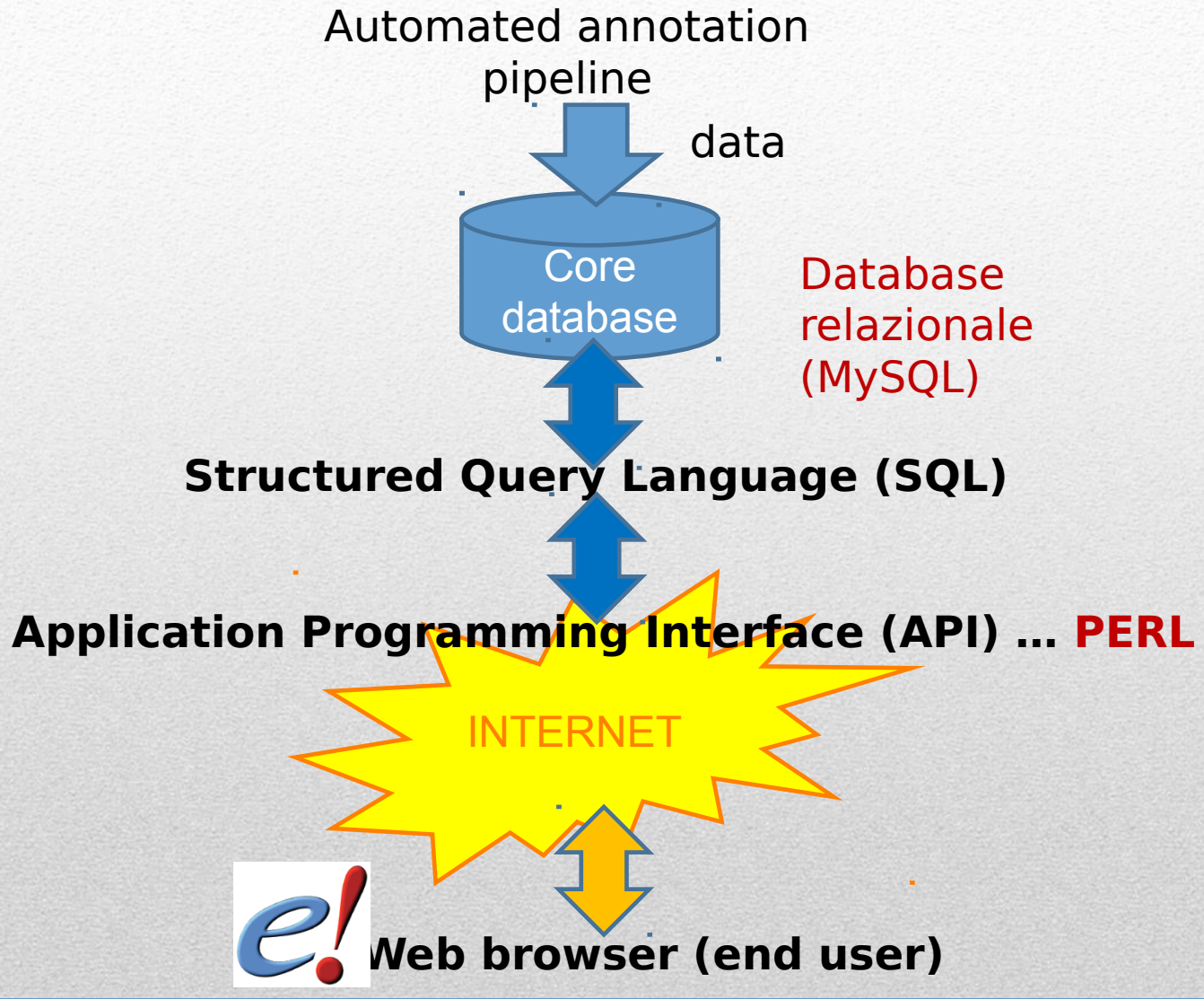


Creazione di «modelli» di geni

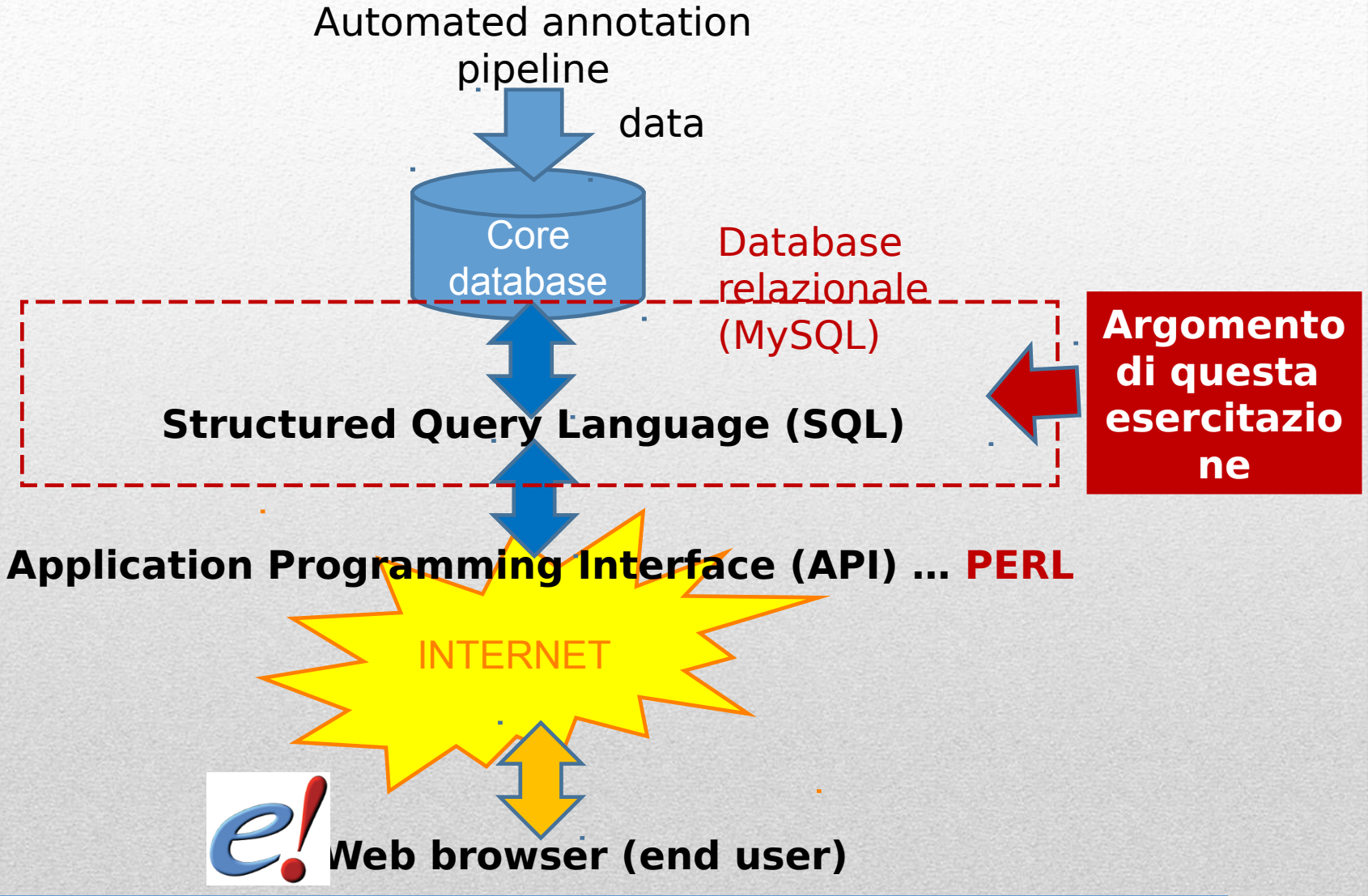


DDT: InfoBiotech supplement

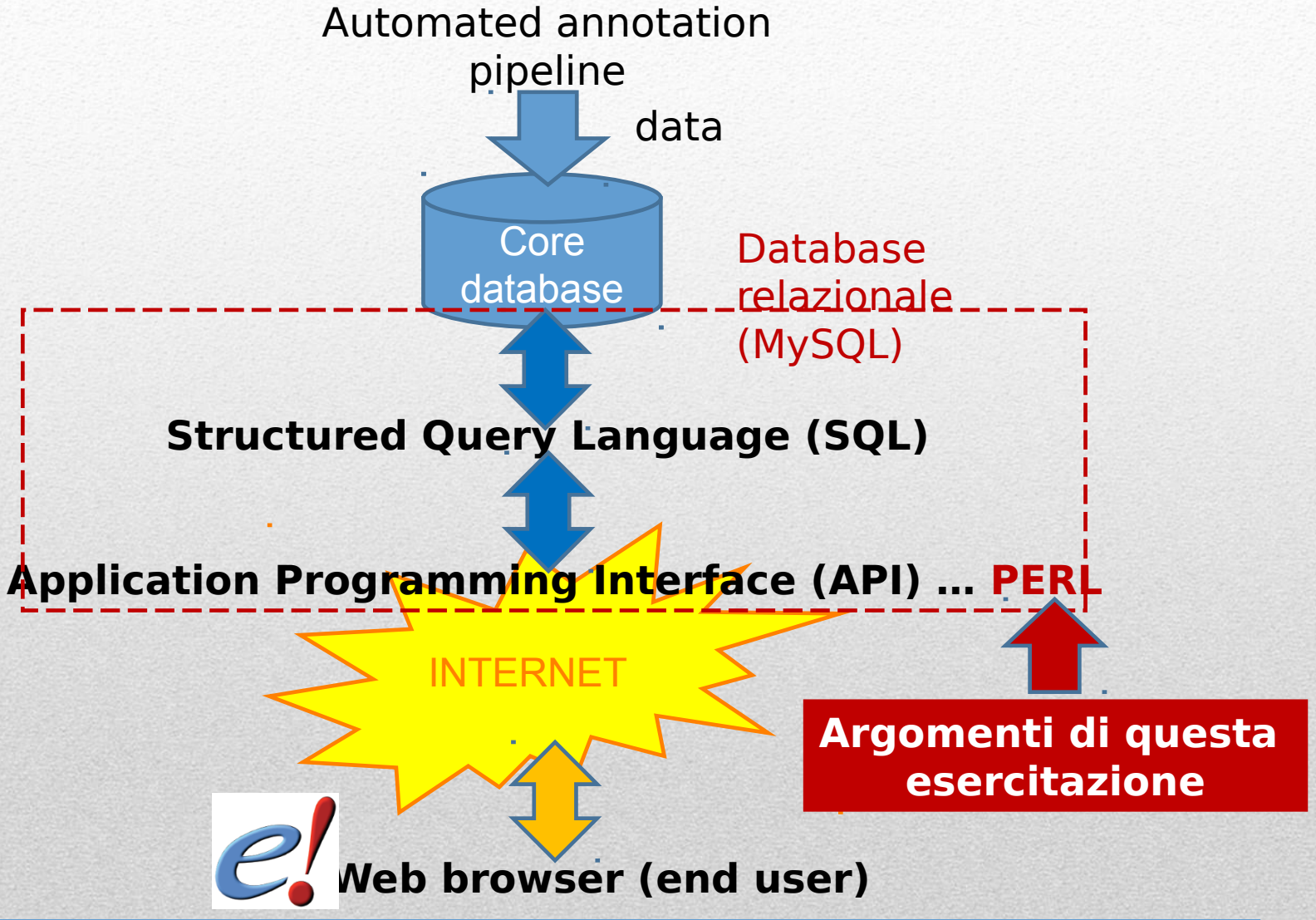
Architettura del browser genomico Ensembl



Architettura del browser genomico Ensembl



Architettura del browser genomico Ensembl



Structured Query Language (SQL) e (R)DBMS

I database sono estremamente eterogenei per quanto riguarda la loro struttura e la quantità di dati contenuta. Essi possono essere costituiti da file di testo ASCII o file che rappresentano complesse strutture composte da alberi binari (ad es. Oracle o Sybase). In ogni caso un database è un contenitore di dati.

PROBLEMA:

Se un database è una semplice collezione di dati … chi tiene traccia del cambiamento dei dati stessi?

Questo è il ruolo dei sistemi di gestione delle basi di dati (database management systems o DBMS). Alcuni DBMS sono relazionali. In tal caso ci si riferisce ad essi come relational DBMS o RDBMS. Le relazioni su cui si basano i sistemi RDBMS assicurano che diverse colelzioni di dati (ad es. tabelle) possano essere interrogate "all'unisono". Le relazioni, di fatto, rappresentano delle regole di integrità referenziale tra collezioni di dati. Supponiamo di avere un RDBMS che contiene I dati di tutti gli impiegati di un azienza e di avere 2 tabelle: reparto e impiegato. Tra di esse potrebbe esistere una relazione che permette l'inserimento di un nuovo impiegato SOLO se esso è assegnato ad un reparto esistente.

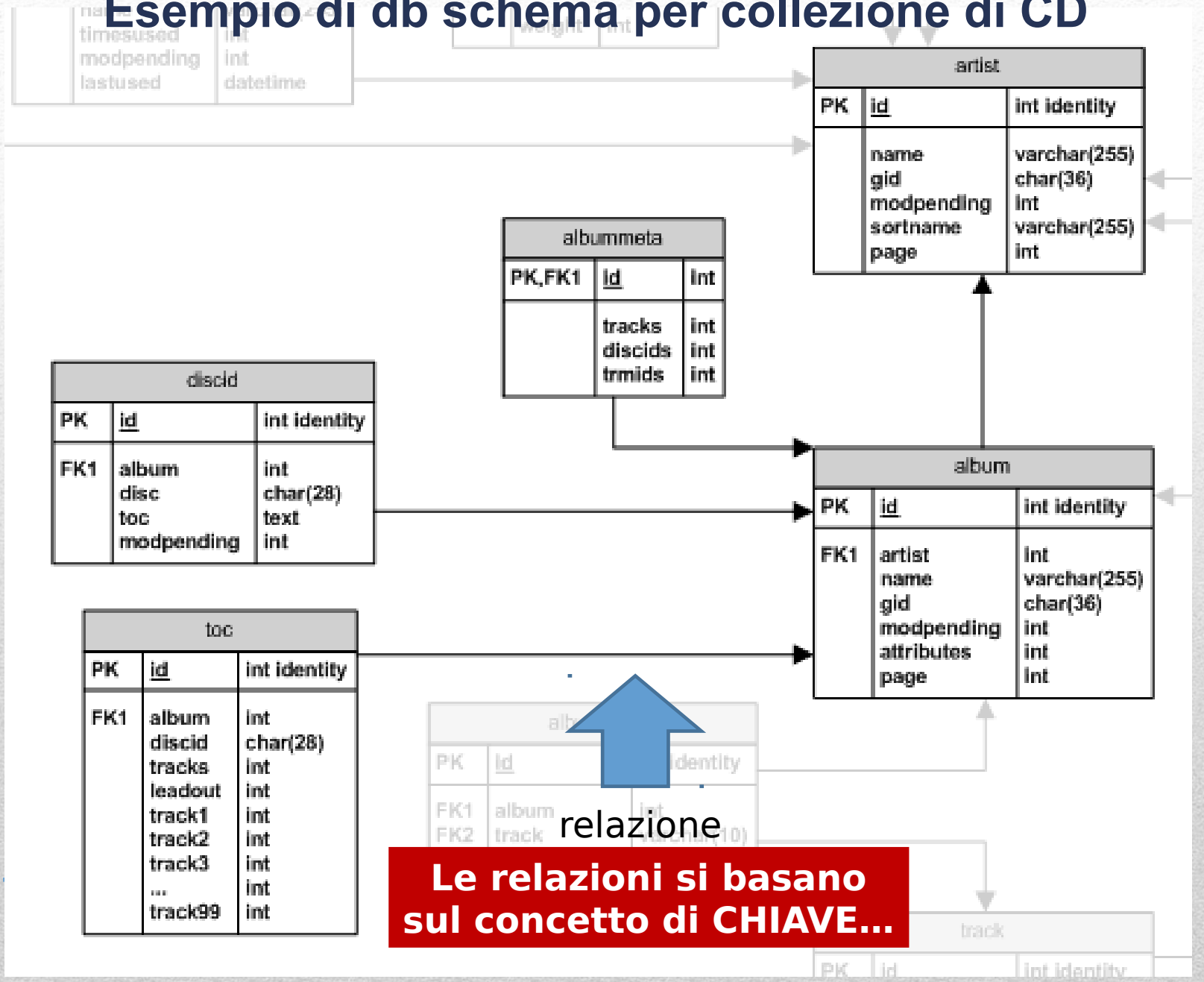
Structured Query Language (SQL) e (R)DBMS

Un database relazionale (come quello associato alla maggioranza delle banche dati genomiche) è costituito da :

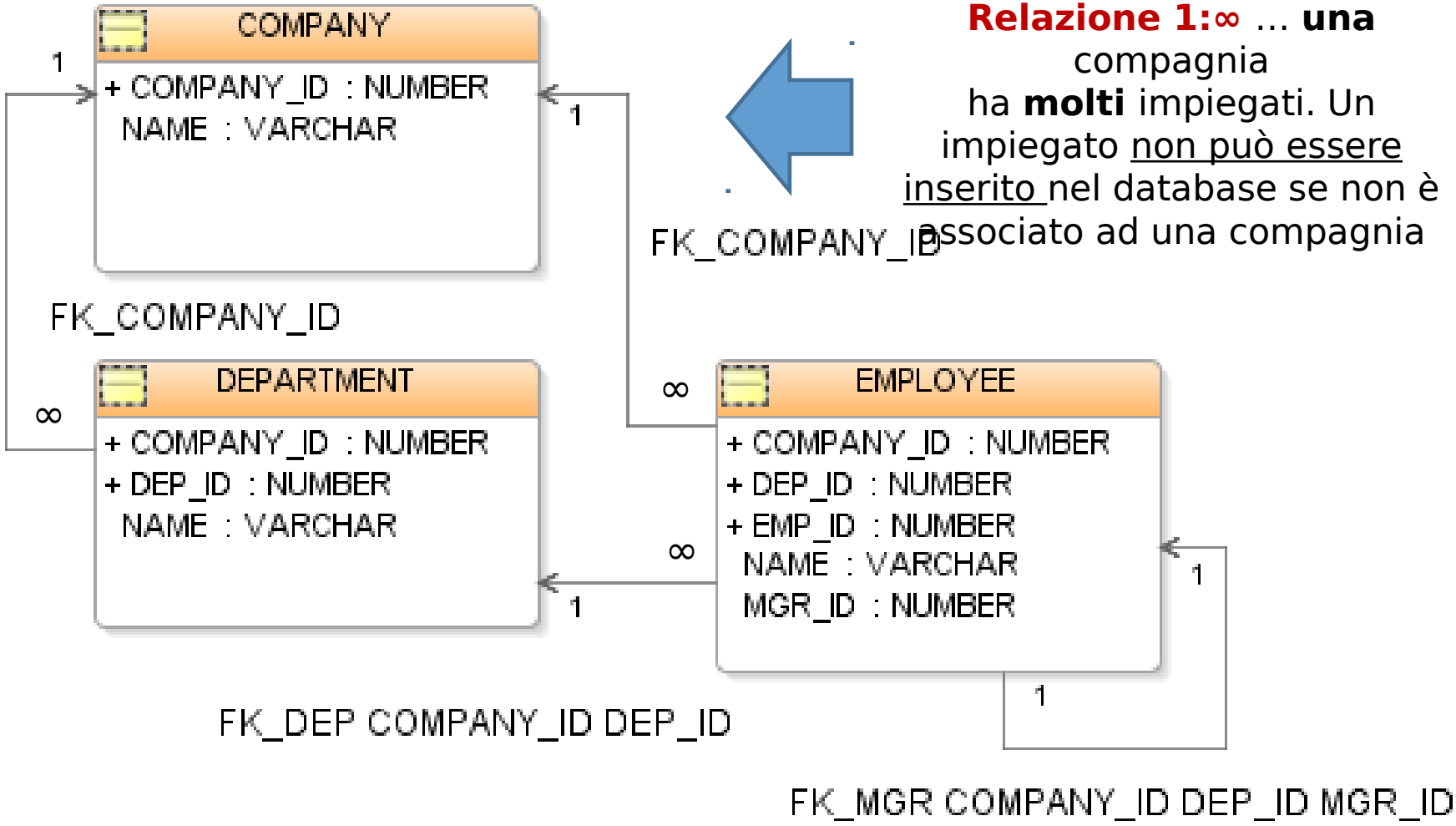
1.Una parte INVARIANTE nel tempo detta database schema. Essa definisce la struttura logica delle unità di memorizzazione delle informazioni. Tale struttura, di solito, viene rappresentata sottoforma di tabella. La rappresentazione tabulare permette di nascondere l dettagli del formato reale di memorizzazione su disco.

1.I dati veri e propri: ad essi ci si riferisce con il termine generico di istanze. Per ogni tabella presente nella banca dati è disponibile una DEFINIZIONE composta da numero e nomi dei campi (colonne) della tabella, tipo di dato ammesso in ogni campo e altre caratteristiche (che descrivono ad esempio, il coinvolgimento di una relazione di integrità associata ad un dato campo). Prima di inserire una nuova riga in una tabella il sistema RDBMS verifica che la colelzione di dati (la riga della tabella) tispetti tutte le specifiche della tabella stessa. Un altro modo comune di riferirsi alle righe delle tabelle è il termine RECORD.

Esempio di db schema per collezione di CD



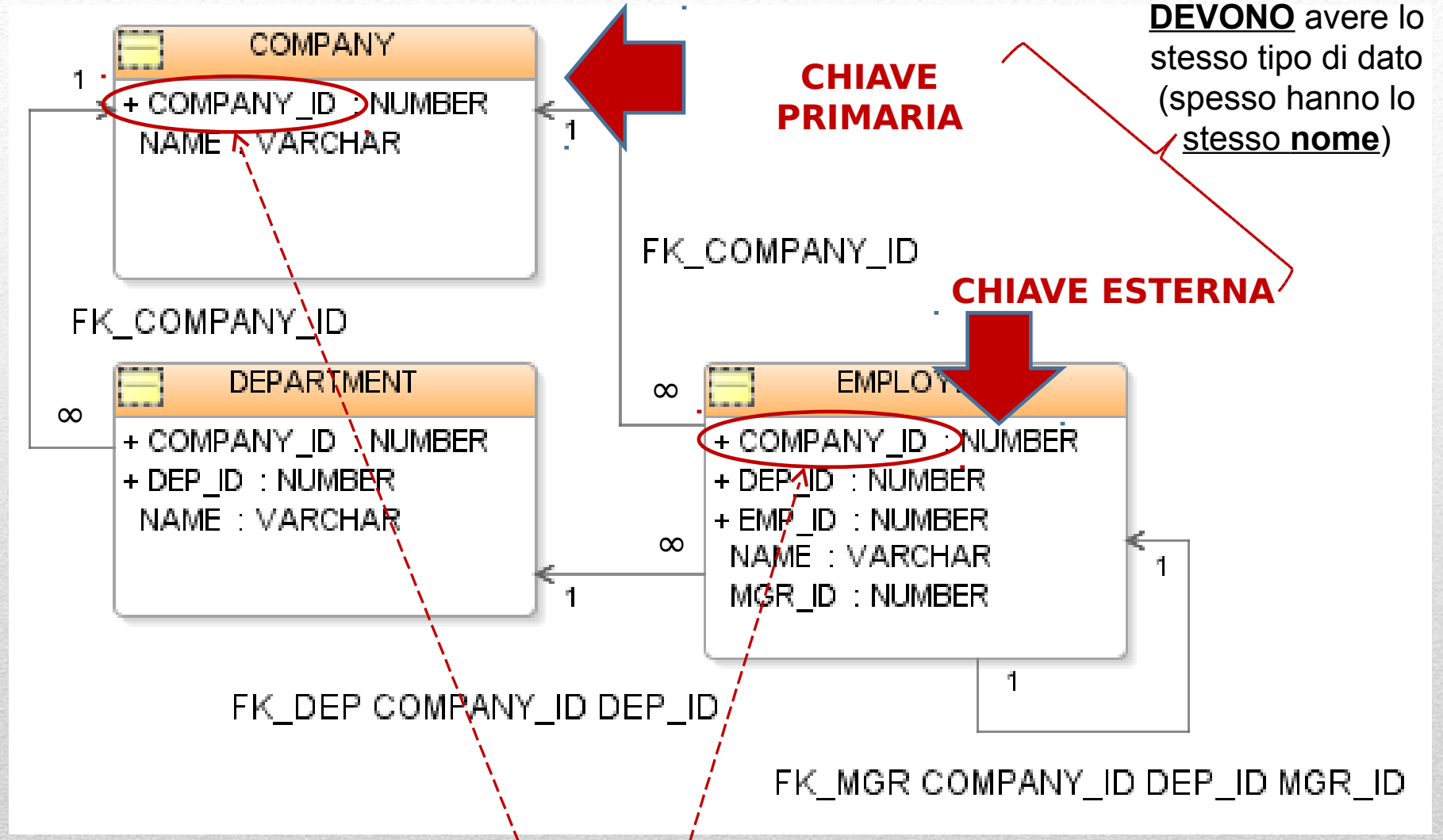
Esempio di db schema per collezione di CD



Relazione 1:∞ … **una** compagnia ha **molti** impiegati. Un impiegato non può essere inserito nel database se non è associato ad una compagnia

Le relazioni si basano sul concetto di CHIAVE…

Chiavi primarie e chiavi esterne



Questa relazione si basa su due campi:
COMPANY_ID (tabella COMPANY) e COMPANY_ID (tabella EMPLOYEE)

Interazione con RDBMS e ruolo di SQL

E' necessaro uno strumento che permetta di interagire con la banca dati. Questo ruolo è svolto da un linguaggio standardizzato detto Structured Query Language (SQL). SQL permette non solo l'estrazione dei dati ma anche la creazione/modifica di database e tabelle nonchè la definizione di vincoli relazionali. SQL si divide in:

- **DATA DEFINITION LANGUAGE (DDL): linguaggio di definizione dei dati, serve per creare databases, definizioni di tabelle e vincoli di integrità referenziale. Permette inoltre di modificare la struttura di tabelle esistenti.**

- **DATA MANIPULATION LANGUAGE (DML): insieme di enunciati che permettono, principalmente, di estrarre informazioni da una banca dati.**

A noi interessa DML (DDL non verrà trattato)

Operazioni realizzabili mediante SQL DML

SQL DML permette di realizzare diverse operazioni che possono essere attribuite a tre grandi macrocategorie:

- **PROIEZIONE:** **Estrazione di attributi (valori contenuti in un sottoinsieme di colonne di una tabella specificate dall'utente)**

- **ESTRAZIONE:** **Selezione di alcune righe (record) da una tabella nel caso in cui queste corrispondano ad alcuni criteri specificati dall'utente**

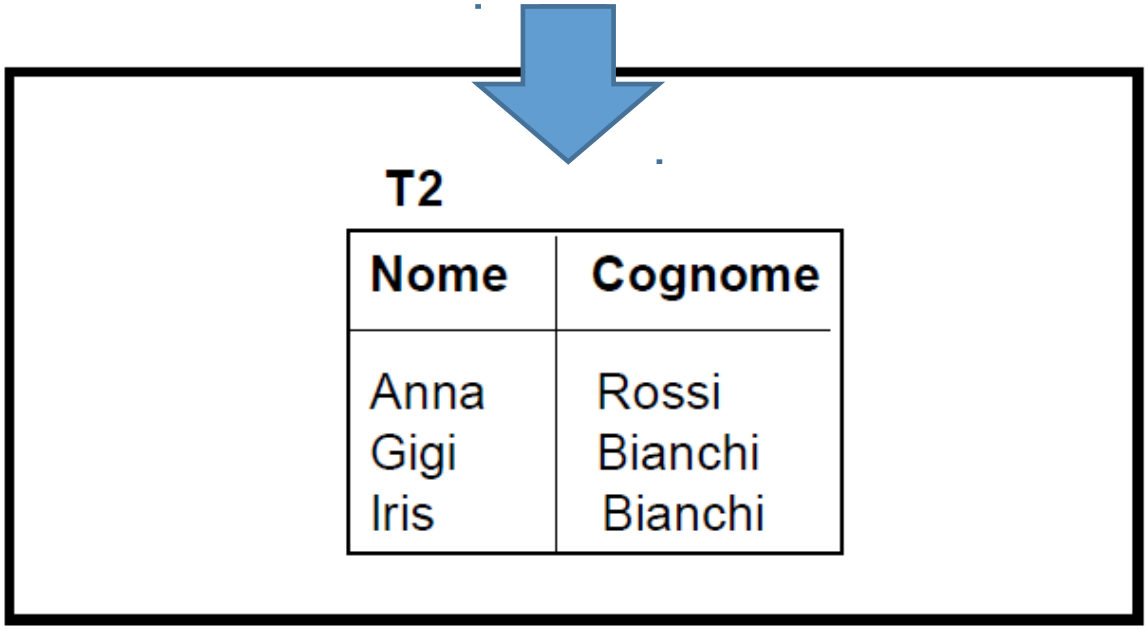
- **JOIN:** **Interrogazione simultanea di più tabelle basata su relazioni. Concettualmente equivale a creare in memoria una macrotabella costituita dai dati contenuti in più tabelle. Solitamente dopo il join viene effettuata un'estrazione.**

Esempio di PROIEZIONE

T1

Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea
Iris	Bianchi	15/9/45	CN

La **proiezione** di T1 sugli attributi Nome e Cognome restituisce



T2

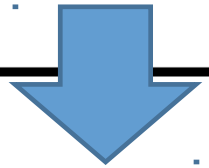
Nome	Cognome
Anna	Rossi
Gigi	Bianchi
Iris	Bianchi

Esempio di ESTRAZIONE (o selezione)

T1

Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea
Iris	Bianchi	15/9/45	CN

La **selezione** dei record di T1 tali che **"Nato il >= 1/1/1960"** restituisce



T2

Nome	Cognome	Nato il	Nato a
Anna	Rossi	2/2/71	TO
Gigi	Bianchi	23/4/80	Ivrea

Esempio di JOIN

Chiave primaria

Chiave esterna

T1

Titolo	Autore	Codice
Poesie	Rossi	111
Prosa	Verdi	222
Elegie	Verdi	333

T2

Utente	Cod libro
Pippo	111
Pippo	222
Pluto	111

Il **join** fra le due tabelle restituisce

NB: nessun utente ha acquistato questo libro

T3

Titolo	Autore	Codice	Utente	Cod libro
Poesie	Rossi	111	Pippo	111
Poesie	Rossi	111	Pluto	111
Prosa	Verdi	222	Pippo	222

Il libro non è presente nei risultati

Estrazione di dati mediante SQL: enunciato SELECT

E' lo strumento **principale** per estrarre records. Ha una struttura **composta da 3 parti**:

1.Nella prima parte vengono specificate le operazioni di **proiezione** (nel senso che specifichiamo **quali** campi (colonne) vogliamo estrarre e di **quali tabelle**)

1.Nella seconda parte viene specificata la **tabella** (o la macrotabella definita mediante una o più operazioni di **join**) da cui vogliamo estrarre i dati

1.Nella terza ed ultima parte è possibile specificare i **criteri di estrazione** ossia l'insieme di regole a cui un record **DEVE** essere conforme perchè venga restituito tra i risultati dell'interrogazione (query) SQL.

Struttura :

questa parte è **opzionale**

SELECT 1 FROM 2 WHERE 3 ;

I nomi dei campi sono separati da ,

WARNING: alcuni programmi richiedono che la stringa di interrogazione SQL termini con **;**

Esempio di utilizzo di enunciato SELECT

Il database **Ensembl core** contiene una tabella **gene**:

Field	Type		Null	Key	Default		Extra
gene_id	int(10) unsigned	16B	NO	PRI	(NULL)	0K	auto_increment
biotype	varchar(40)	11B	NO		(NULL)	0K	
analysis_id	smallint(5) unsigned	20B	NO	MUL	(NULL)	0K	
seq_region_id	int(10) unsigned	16B	NO	MUL	(NULL)	0K	
seq_region_start	int(10) unsigned	16B	NO		(NULL)	0K	
seq_region_end	int(10) unsigned	16B	NO		(NULL)	0K	
seq_region_strand	tinyint(2)	10B	NO		(NULL)	0K	
display_xref_id	int(10) unsigned	16B	YES	MUL	(NULL)	0K	
source	varchar(20)	11B	NO		(NULL)	0K	
status	enum('KNOWN','NOVEL','PUTATIVE','PREDICTED','KNOWN_BY_PRO...	76B	YES		(NULL)	0K	
description	text	4B	YES		(NULL)	0K	
is_current	tinyint(1)	10B	NO		1	1B	
canonical_transcript_id	int(10) unsigned	16B	NO		(NULL)	0K	
canonical_annotation	varchar(255)	12B	YES		(NULL)	0K	

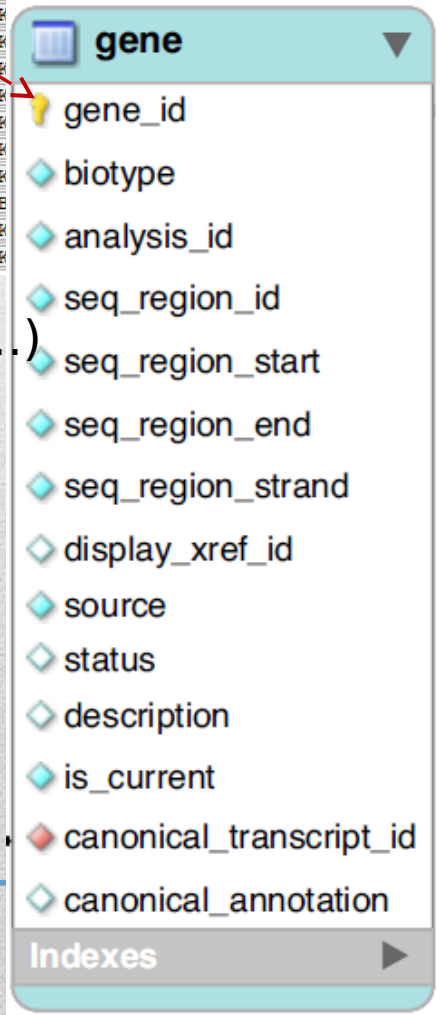
chiave primaria

chiavi esterne

descrizione dettagliata (nomi campi, tipi di dato …)

SQL: DESCRIBE gene;

Rappresentazione **semplificata** (nomi campo + simboli ma non tipo di dato). Comune in molti strumenti ad interfaccia grafica ed **estremamente comune** nei diagrammi che descrivono gli schemi delle banche dati



Esempio di utilizzo di enunciato SELECT

Conoscendo la definizione (struttura) della tabella(e) a cui siamo interessati possiamo scrivere la query SQL per estrarre dati da essa(e):

proiezione

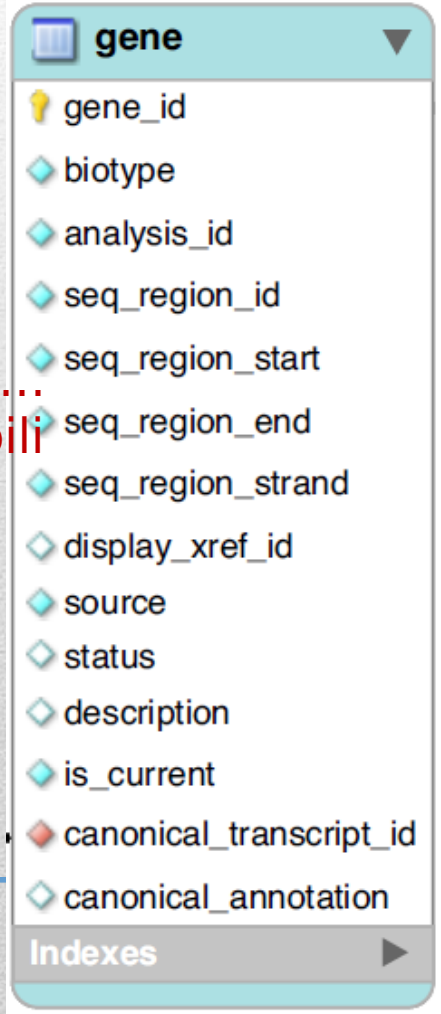
SELECT gene_id, biotype, status FROM gene;

Nessuna proiezione: estrae tutti i campi

Nessun criterio di selezione … estrae tutti i record disponibili

SELECT * FROM gene WHERE biotype = 'protein_coding';

Criterio di selezione: estrae **solo** i geni che codificano per proteine



Strumenti free per l'accesso a banche dati relazionali

Proveremo ad effettuare alcuni esperimenti pratici utilizzando uno strumento free: **MySQL workbench**

Scaricatelo da questo sito:
https://dev.mysql.com/downloads/workbench/
(scaricate l'ultima versione)

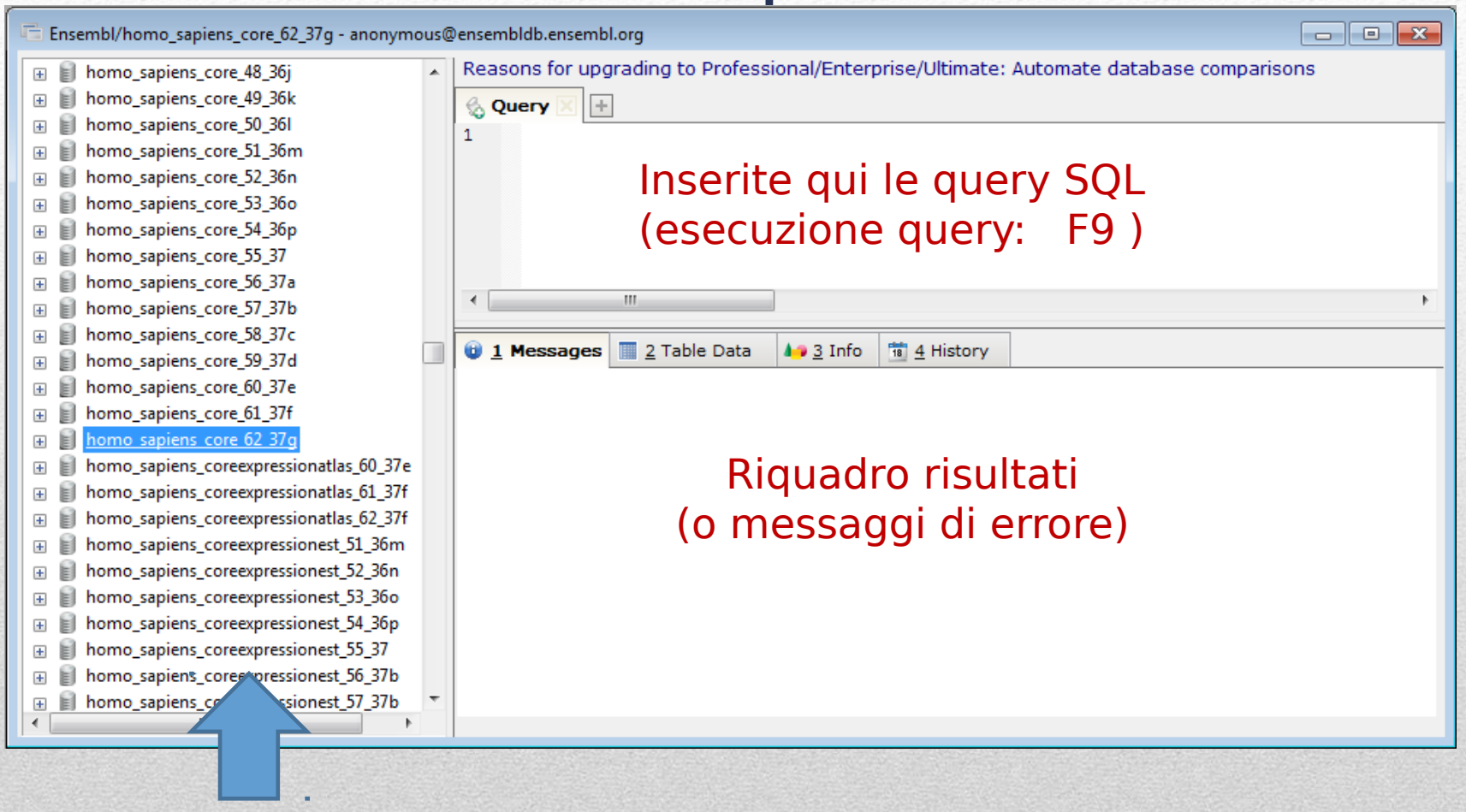
Provate ad installarlo in una directory in cui **avete i permessi di scrittura**.

Una volta installato definite i parametri per una nuova connessione:

File -> New **connection**
Valori:
Nome connessione EnsEMBL
MySQL Host Address ensembldb.ensembl.org
Username anonymous
Port 5306
(lasciate vuota la password che non serve)

Accesso in sola lettura

Alcuni elementi di finestra principale di un client sql

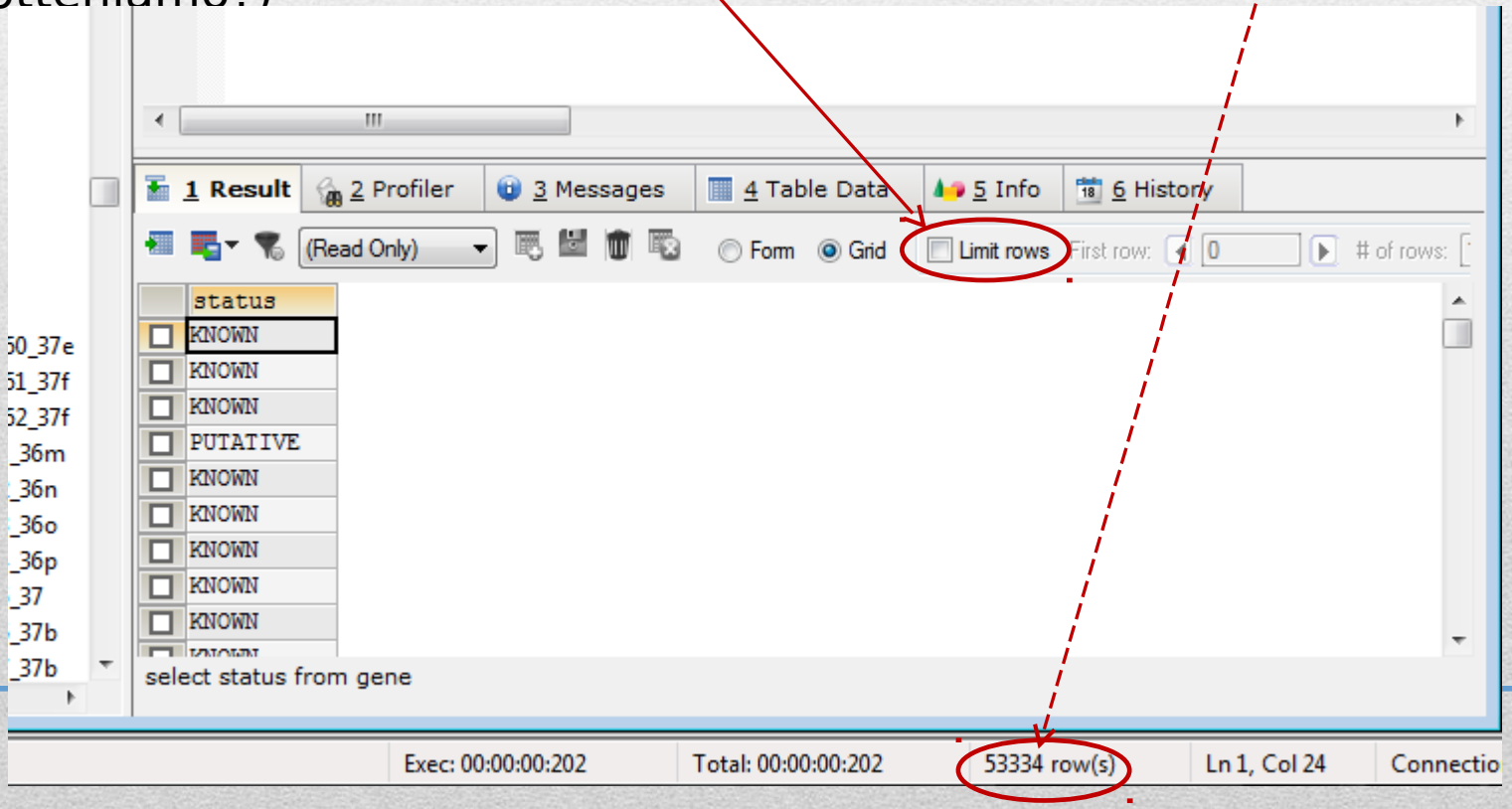


Database disponibili: a noi interessa **homo_sapiens_core_62_37g**
(click sx per selezionarlo)

Interrogazione diretta di Ensembl

Proviamo ad utilizzare le seguenti query SQL:

1.SELECT * FROM gene LIMIT 100;
2.SELECT status FROM gene; (1000 record)
3.Deselezioniamo la casella ‘Limit rows’ (quanti record otteniamo?)

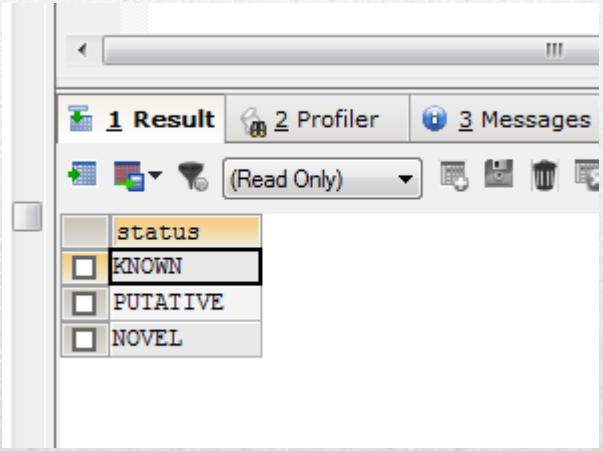


Interrogazione diretta di Ensembl

Quali sono i possibili valori presenti un una data colonna?

SELECT **DISTINCT(**status**)** FROM gene;

Quanti record ottenete?



NON utilizzeremo SQLyog per realizzare i nostri accessi diretti ad Ensembl (utilizzeremo Perl), ma esso è uno strumento molto comodo per testare le query SQL **prima** di inserirle in uno script, in modo da essere sicuri che si comportino secondo le attese.

Esecuzione di query SQL da remoto in Perl

```
# PERL MODULES
use DBI;
use DBD::mysql;

# CONFIG VARIABLES
$platform = "mysql";
$host = "ensembldb.ensembl.org";
$port = "5306";
$user = "anonymous";
$pw = "";
$database="homo_sapiens_core_62_37g";

#DATA SOURCE NAME ##############################
$dsn = "dbi:mysql:$database:$host:5306";

#CONNECTION ####################################
$DBIconnect = DBI->connect($dsn, $user, $pw);

#Query #########################################

$sqlquery = "select * from gene limit 10";

$sth = $DBIconnect->prepare($sqlquery);

$sth->execute;

#PRINT RESULTS #################################
while (@row = $sth->fetchrow_array) {
print "@row\n";
}
```

Librerie SQL

Parametri di connessione

Connessione

Interrogazione

Stampa risultati

Esecuzione di query SQL da remoto in Perl

```
# PERL MODULES
use DBI;
use DBD::mysql;

# CONFIG VARIABLES
$platform = "mysql";
$host = "ensembldb.ensembl.org";
$port = "5
$user = "a
$pw = "";
$database="

#DATA SOUR
$dsn = "db

#CONNECTIO
$DBIconnec

#Query ###

$sqlquery

$sth = $DB

$sth->execute;

#PRINT RESULTS ###################################
while (@row = $sth->fetchrow_array) {
print "@row\n";
}
```

Librerie SQL

Parametri di connessione

ne

one

Stampa risultati

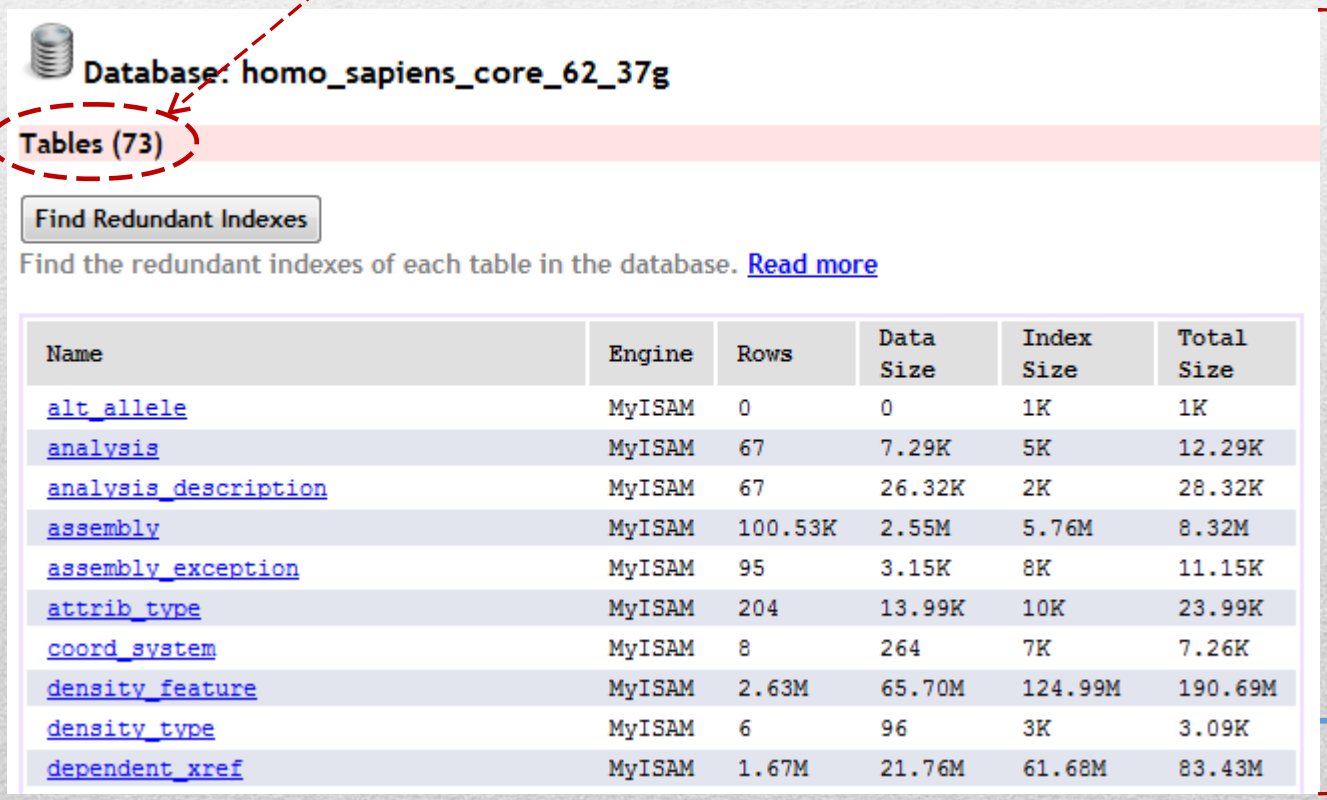
NB: prima di poter utilizzare questo script e' necessario installare alcune librerie aggiuntive in Perl. I loro nomi sono DBI e DBD-mysql e si installano da riga di comando come segue:

ppm install dbi
ppm install DBD-mysql

Esecuzione di query SQL da remoto in Perl

Sembra tutto relativamente semplice … quindi dov'è la difficoltà? La difficoltà sta nella costruzione delle stringhe che contengono le query SQL. Noi abbiamo visto un esempio che interroga **1 tabella** del database core di Ensembl dedicato ad homo sapiens.

Ma **quante tabelle contiene** questo database?



Database: homo_sapiens_core_62_37g

Tables (73)

Find Redundant Indexes

Find the redundant indexes of each table in the database. Read more

Name	Engine	Rows	Data Size	Index Size	Total Size
alt_allele	MyISAM	0	0	1K	1K
analysis	MyISAM	67	7.29K	5K	12.29K
analysis_description	MyISAM	67	26.32K	2K	28.32K
assembly	MyISAM	100.53K	2.55M	5.76M	8.32M
assembly_exception	MyISAM	95	3.15K	8K	11.15K
attrib_type	MyISAM	204	13.99K	10K	23.99K
coord_system	MyISAM	8	264	7K	7.26K
density_feature	MyISAM	2.63M	65.70M	124.99M	190.69M
density_type	MyISAM	6	96	3K	3.09K
dependent_xref	MyISAM	1.67M	21.76M	61.68M	83.43M

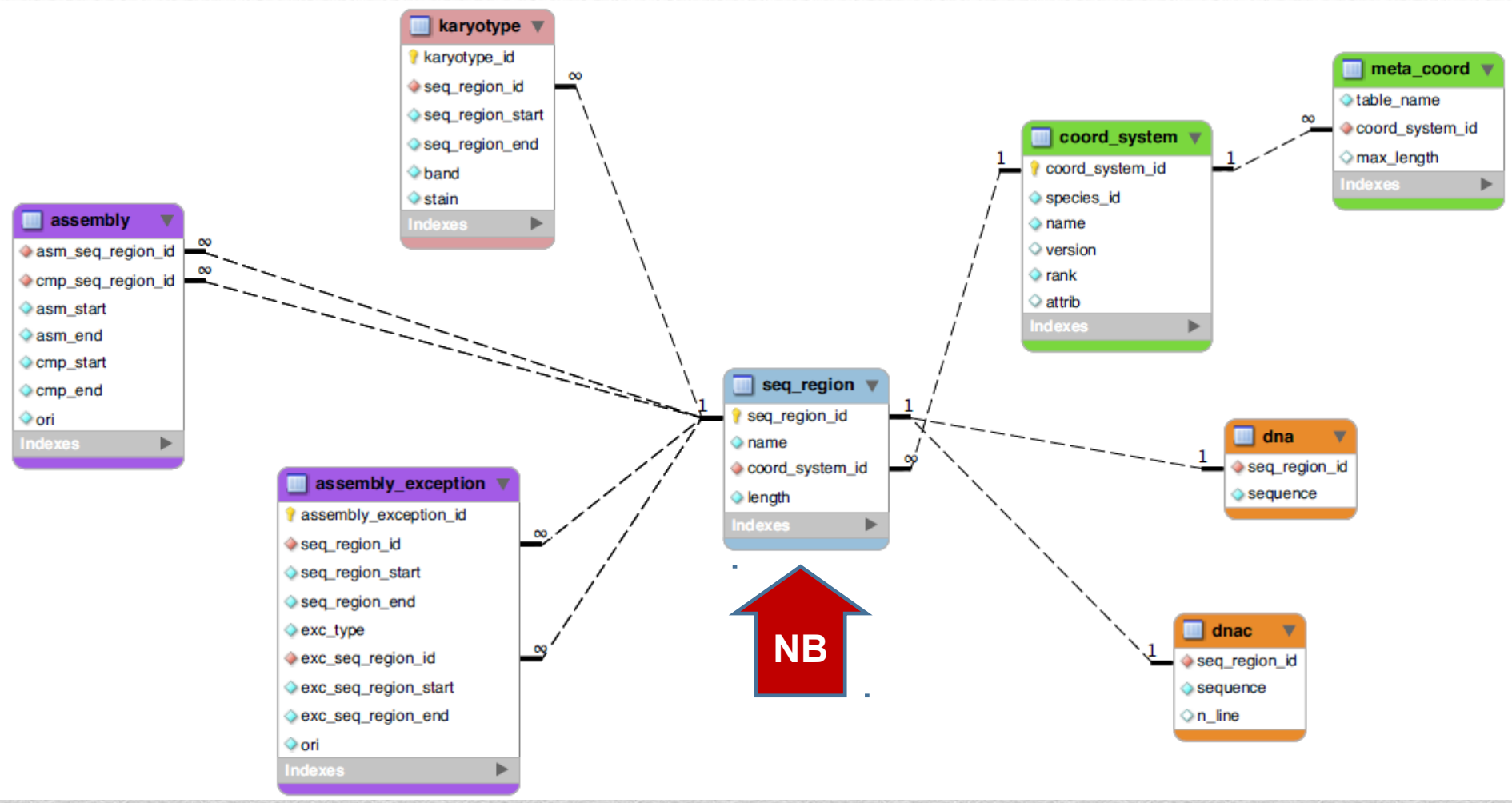
…

Reazioni comuni:

I)Sono troppe …
II)Potevano costruire un database meno complesso
III)AIUTO!!! MI SERVE UNA MAPPA

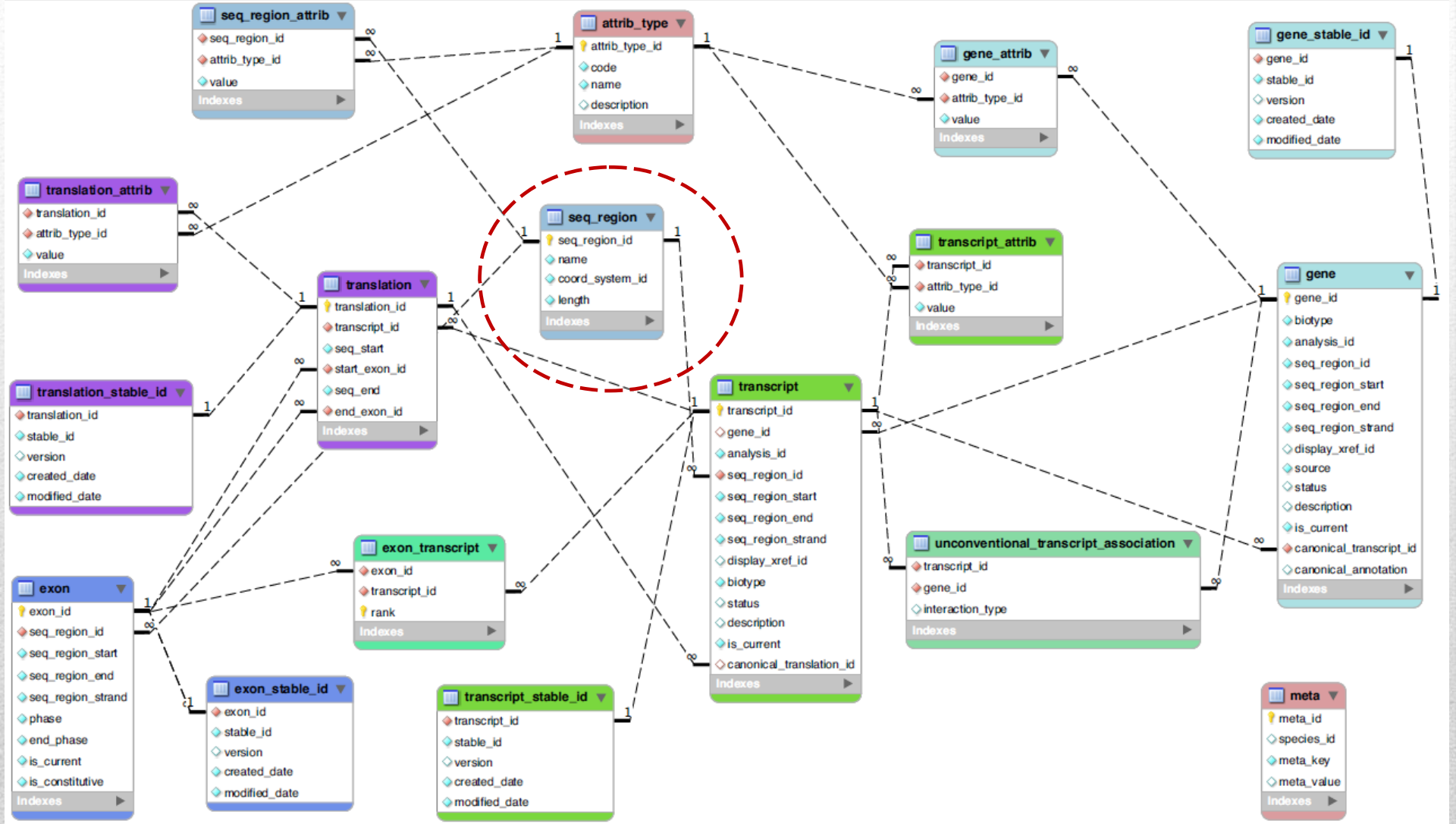
Database schema

Ensembl core db schema (I)



NB: il processo di annotazione genomica **non viene effettuato <u>unicamente</u> sulle sequenze genomiche assemblate**. Parte di esso viene effettuato su cloni, contigui, supercontigui ecc.. OGNI ANNOTAZIONE esiste in uno specifico sistema di coordinate

Ensembl core db schema (II)



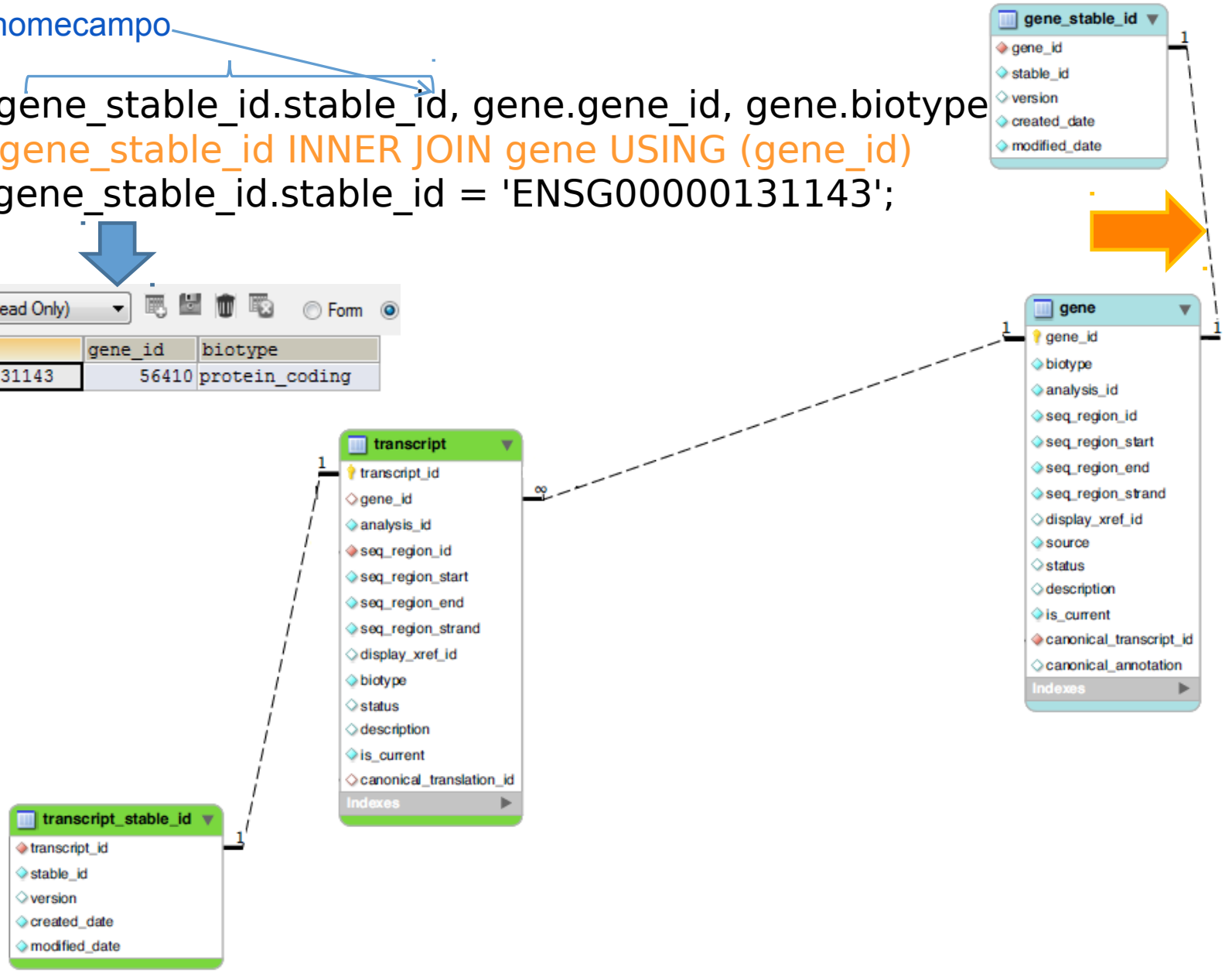
Ensembl core db schema (II) : JOIN

nometabella.nomecampo

```
SELECT gene_stable_id.stable_id, gene.gene_id, gene.biotype
FROM   gene_stable_id INNER JOIN gene USING (gene_id)
WHERE gene_stable_id.stable_id = 'ENSG00000131143';
```

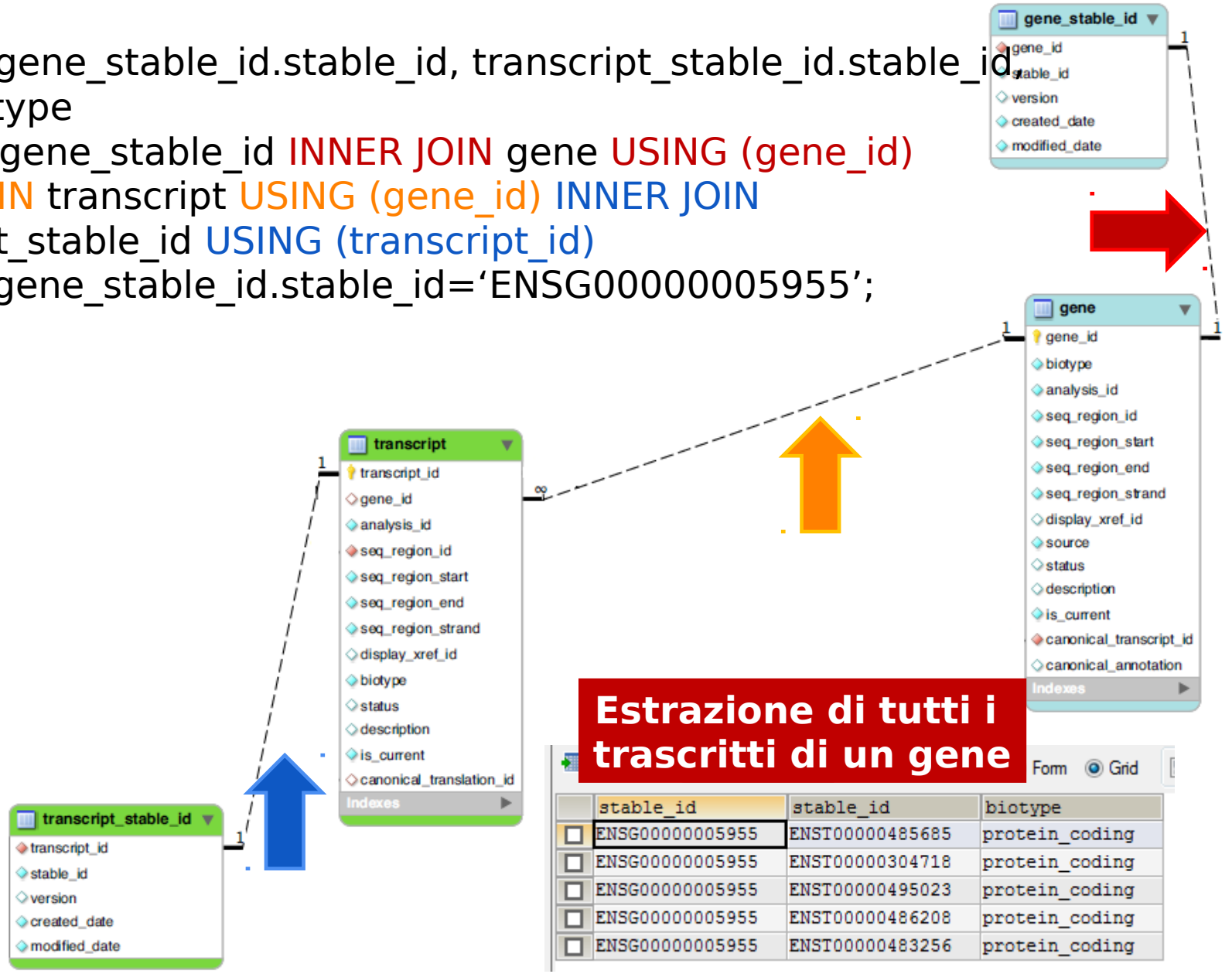
(Read Only) Form

stable_id	gene_id	biotype
ENSG00000131143	56410	protein_coding



Ensembl core db schema (II) : JOIN

SELECT gene_stable_id.stable_id, transcript_stable_id.stable_id, gene.biotype
FROM gene_stable_id INNER JOIN gene USING (gene_id) INNER JOIN transcript USING (gene_id) INNER JOIN transcript_stable_id USING (transcript_id)
WHERE gene_stable_id.stable_id='ENSG00000005955';



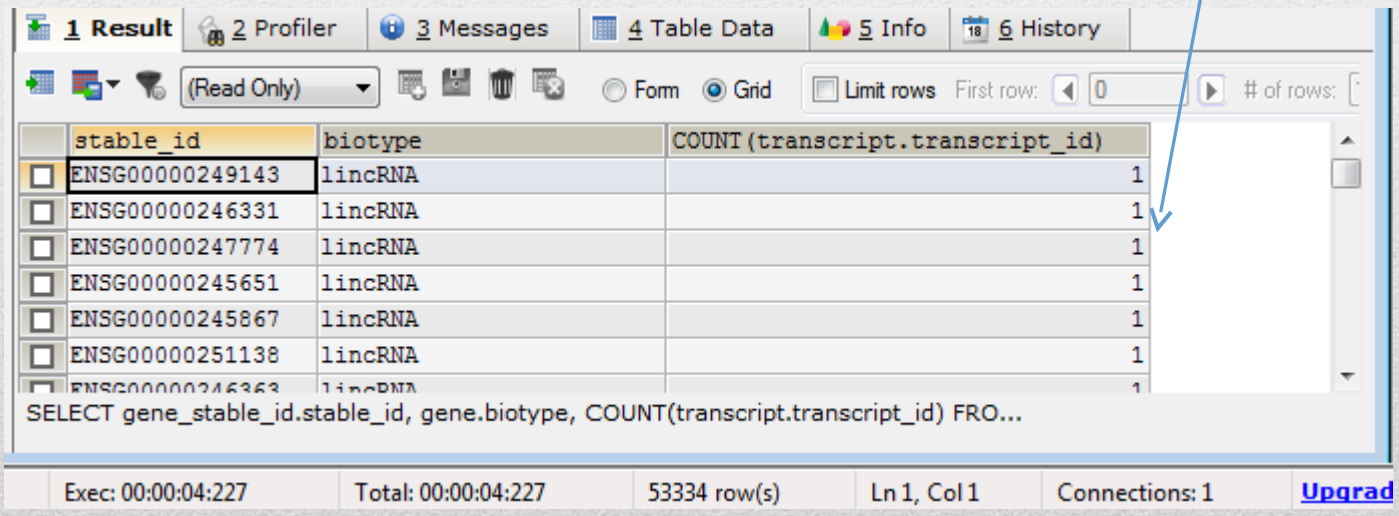
stable_id	stable_id	biotype
ENSG00000005955	ENST00000485685	protein_coding
ENSG00000005955	ENST00000304718	protein_coding
ENSG00000005955	ENST00000495023	protein_coding
ENSG00000005955	ENST00000486208	protein_coding
ENSG00000005955	ENST00000483256	protein_coding

Ensembl : AGGREGAZIONE

```
SELECT gene_stable_id.stable_id, gene.biotype,
COUNT(transcript.transcript_id)
FROM    gene_stable_id INNER JOIN gene USING (gene_id)
INNER JOIN transcript USING (gene_id) GROUP BY
gene_stable_id.stable_id ORDER BY COUNT(transcript.transcript_id);
```

CONTEGGIO di tutti i trascritti di OGNI gene

ordinamento crescente

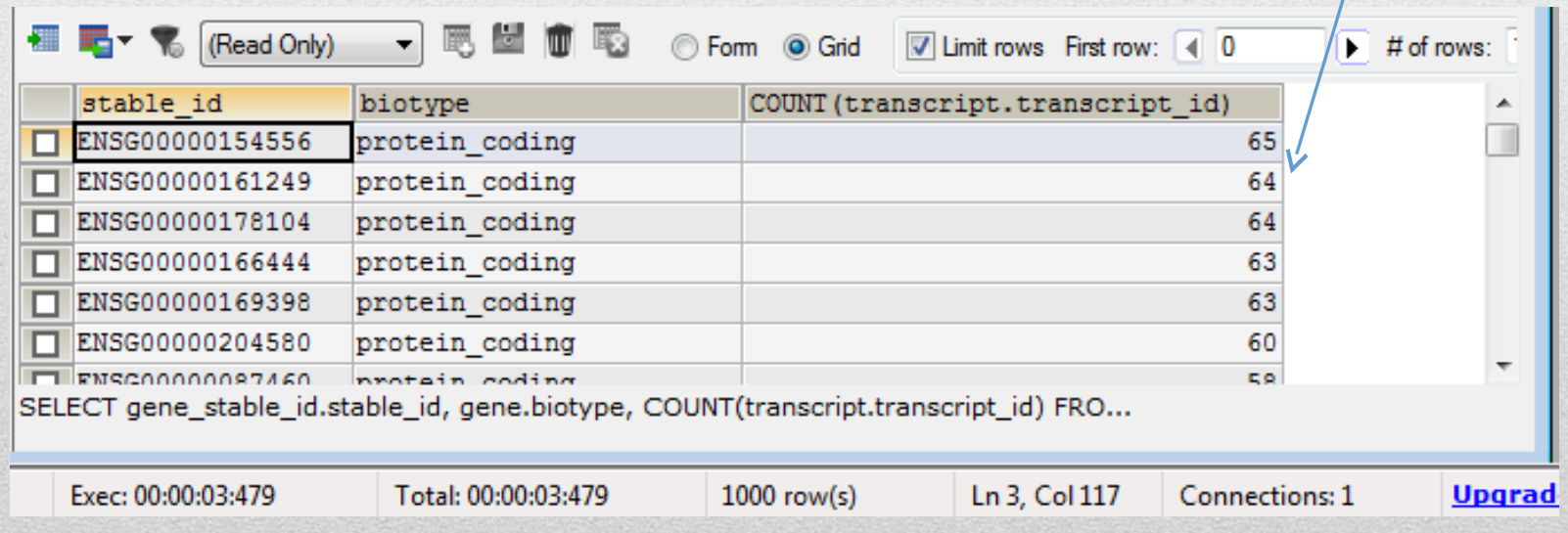


Ensembl : AGGREGAZIONE

```
SELECT gene_stable_id.stable_id, gene.biotype,
COUNT(transcript.transcript_id)
FROM    gene_stable_id INNER JOIN gene USING (gene_id)
INNER JOIN transcript USING (gene_id) GROUP BY
gene_stable_id.stable_id ORDER BY COUNT(transcript.transcript_id)
```

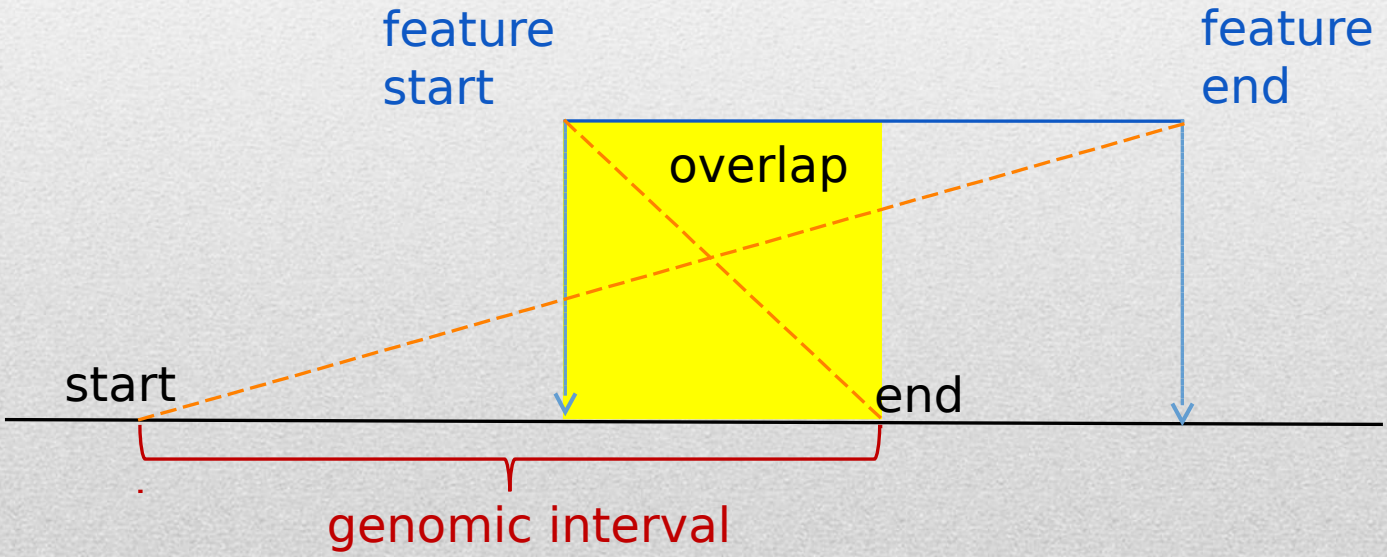
CONTEGGIO di tutti i trascritti di OGNI gene

ordinamento **DE**crescente

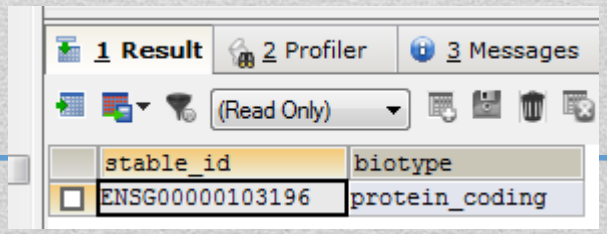


Ensembl core db schema (II) : estrazione su base *posizionale*

```
SELECT gene_stable_id.stable_id, gene.biotype
FROM    seq_region INNER JOIN gene USING (seq_region_id) INNER JOIN
gene_stable_id USING (gene_id)
WHERE NOT(gene.seq_region_start>84966302 OR
gene.seq_region_end<84826528) AND seq_region.name = '16';
```

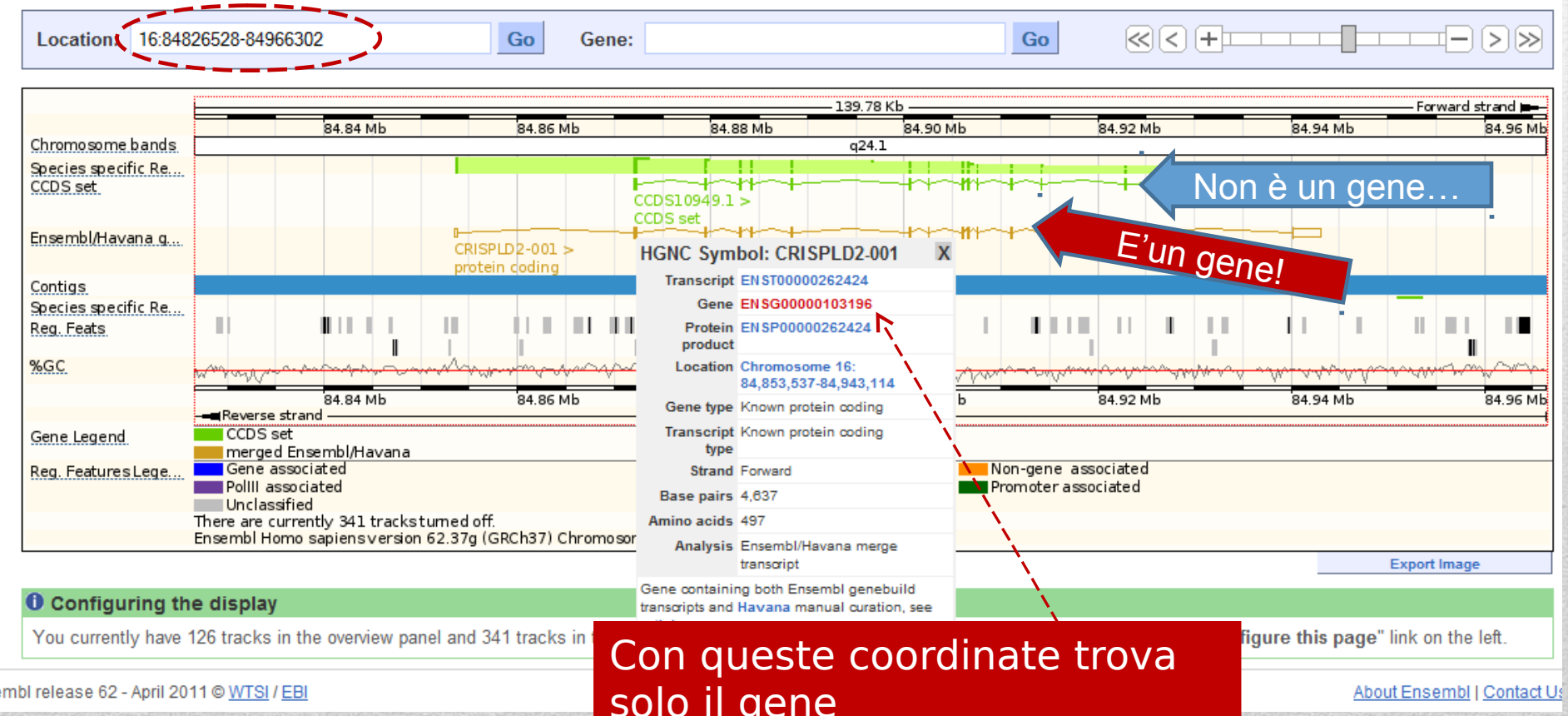


Con queste coordinate trova solo il gene **ENSG00000103196** … sarà vero?



Ensembl core db schema (II) : estrazione su base *posizionale*

```
SELECT gene_stable_id.stable_id, gene.biotype
FROM   seq_region INNER JOIN gene USING (seq_region_id) INNER JOIN
gene_stable_id USING (gene_id)
WHERE NOT(gene.seq_region_start>84966302 OR gene.seq_region_end<84826528) AND
seq_region.name = '16';
```



Ensembl SQL :
estrazione basata su pattern di testo

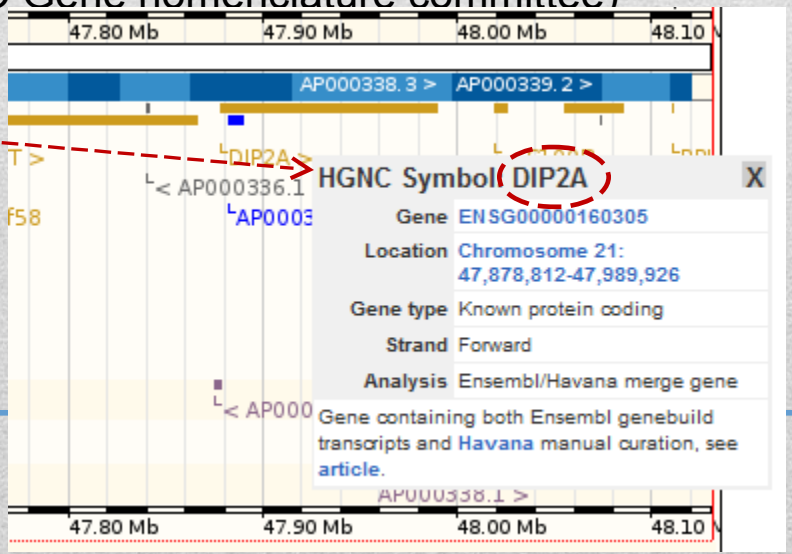
SELECT gene_stable_id.stable_id, gene.biotype, xref.display_label **FROM** gene_stable_id INNER JOIN gene USING (gene_id) INNER JOIN xref ON (gene.display_xref_id = xref.xref_id) **WHERE** xref.display_label **LIKE 'DIP2%'**;

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only) Form Grid

stable_id	biotype	display_label
ENSG00000160305	protein_coding	DIP2A
ENSG00000066084	protein_coding	DIP2B
ENSG00000151240	protein_coding	DIP2C

NBB: Tabella **xref** (**external reference**) fa riferimento ad un dato mappato sul genoma ma **ORIGINARIO** di un'altra banca dati. Nel caso (ad es.) di **DIP2A** si tratta di un gene symbol definito da **HGNC** (HUGO Gene nomenclature committee)



Esercizi (SQL)

- Scrivete una query SQL che restituisca tutti i trascritti di un gene a vostra scelta (**3 pt**)
- Scrivete una query che restituisca **IL NUMERO** degli pseudogeni umani annotati in Ensembl (**3 pt**)
- Scrivete una query che restituisca tutti i geni del cromosoma 1 di tipo **diverso** da protein_coding (**3 pt**)
- Scrivete **uno script** a cui passare come parametro il nome di un gene e che restituisca il numero dei suoi trascritti e, per ciascun trascritto, I nomi e le posizioni dei suoi esoni . Potete realizzare l'esercizio mediante **più query** successive (gene → trascritti, per ogni trascritto → esoni) (6 **pt**).
- Scrivete **uno script** a cui passare come parametro delle coordinate genomiche e che restituisca le simple_feature annotate nella regione genomica, la loro posizione ed il loro tipo . NB: questo esercizio **richiede l'utilizzo di INNER JOIN da una tabella che dovete identificare ad altre due tabelle: seq_region e analysis** (**6 pt**).