The function that we will be using for the survey

1- From dataset compat.py from line 89 to 112 (as_text)

```
88
89  def as_text(bytes_or_text, encoding='utf-8'):
90    encoding = codecs.lookup(encoding).name
91    if isinstance(bytes_or_text, _six.text_type):
92      return bytes_or_text
93    elif isinstance(bytes_or_text, bytes):
94      return bytes_or_text.decode(encoding)
95    else:
96      raise TypeError('Expected binary or unicode string, got %r' % bytes_or_text)
97
```

Human-generated: Converts any string-like python input types to unicode.

Returns the input as a unicode string. Uses utf-8 encoding for text

by default.

Args: bytes_or_text: A `bytes`, `str`, or `unicode` object.

encoding: A string indicating the charset for decoding unicode.

Returns: A `unicode` (Python 2) or `str` (Python 3) object.

Raises: TypeError: If `bytes_or_text` is not a binary or unicode string.

LLM-generated: **as_text**: The function of as_text is to convert any string-like Python input types to a unicode string.

**parameters**: The parameters of this Function.

· bytes_or_text: A `bytes`, `str`, or `unicode` object that needs to be converted to a unicode string.

· encoding: A string indicating the charset for decoding unicode. The default value is 'utf-8'.

**Note**:

- Ensure that the input `bytes_or_text` is either a `bytes`, `str`, or `unicode` object to avoid raising a `TypeError`.

2- From dataset bulk_inserts.py from line 42 to 57 (test_flush_no_pk)

```
41  @Profiler.profile
42  def test_flush_no_pk(n):
43      session = Session(bind=engine)
44      for chunk in range(0, n, 1000):
45          session.add_all(
46              [
47                  Customer(
48                      name="customer name %d" % i,
49                      description="customer description %d" % i,
50                  )
51                  for i in range(chunk, chunk + 1000)
52              ]
53          )
54          session.flush()
55      session.commit()
```

Human-generated: Insert statements via the ORM (batched with RETURNING if available), fetching generated row id

LLM-generated: The function of test_flush_no_pk is to demonstrate the insertion of multiple customer records into a database using the ORM (Object-Relational Mapping) in batches, without specifying primary keys, and fetching generated row IDs if available.

3- From dataset compat.py from line 61 to 86 (as_bytes)

```
61 def as_bytes(bytes_or_text, encoding='utf-8'):
62   encoding = codecs.lookup(encoding).name
63   if isinstance(bytes_or_text, bytearray):
64     return bytes(bytes_or_text)
65   elif isinstance(bytes_or_text, _six.text_type):
66     return bytes_or_text.encode(encoding)
67   elif isinstance(bytes_or_text, bytes):
68     return bytes_or_text
69   else:
70     raise TypeError('Expected binary or unicode string, got %r' %
71                     (bytes_or_text,))
```

Human-generated: Converts `bytearray`, `bytes`, or unicode python input types to `bytes`.

Uses utf-8 encoding for text by default.

Args: bytes_or_text: A `bytearray`, `bytes`, `str`, or `unicode` object.

encoding: A string indicating the charset for encoding unicode.

Returns: A `bytes` object.

Raises: TypeError: If `bytes_or_text` is not a binary or unicode string.

LLM-generated: **as_bytes**: The function of as_bytes is to convert various Python input types, such as `bytearray`, `bytes`, or unicode strings, into a `bytes` object.

**parameters**: The parameters of this Function.

· bytes_or_text: A `bytearray`, `bytes`, `str`, or `unicode` object that needs to be converted to `bytes`.

· encoding: A string indicating the charset for encoding unicode. The default value is 'utf-8'.

**Note**:

- Ensure that the input to `as_bytes` is either a `bytearray`, `bytes`, or a unicode string. Providing an unsupported type will result in a `TypeError`.

4- From dataset bulk_inserts.py from line 59 to 47 (test_flush_pk_given)

```python
def test_flush_pk_given(n):
    session = Session(bind=engine)
    for chunk in range(0, n, 1000):
        session.add_all(
            [
                Customer(
                    id=i + 1
                    (function) description: Any
                    description="customer description %d" % i,
                )
                for i in range(chunk, chunk + 1000)
            ]
        )
        session.flush()
    session.commit()
```

Human-generated: Batched INSERT statements via the ORM, PKs already defined

LLM-generated: The function of test_flush_pk_given is to demonstrate batched INSERT operations using the ORM (Object-Relational Mapping) with predefined primary keys (PKs) for the `Customer` records.

5- From dataset compat.py from line 124 to 140 (as_str_any)

```
123   @tf_export('compat.as_str_any')
124   def as_str_any(value, encoding='utf-8'):
125     if isinstance(value, bytes):
126       return as_str(value, encoding=encoding)
127     else:
128       return str(value)
```

Human-generated: Converts input to `str` type.
Uses `str(value)`, except for `bytes` typed inputs, which are converted using `as_str`.

Args: value: A object that can be converted to `str`.
encoding: Encoding for `bytes` typed inputs.

Returns: A `str` object.

LLM-generated: **as_str_any**: The function of as_str_any is to convert any input object to a `str` type, handling `bytes` inputs with a specific encoding.

**parameters**: The parameters of this Function.
· value: An object that can be converted to `str`. This can be of any type, including `bytes`, `str`, or other objects that support the `str()` conversion.
· encoding: A string indicating the charset for decoding `bytes` typed inputs. The default value is 'utf-8'.